Khalid Al-Begain
Dieter Fiems
Gábor Horváth (Eds.)

LNCS 5513

Analytical and Stochastic Modeling Techniques and Applications

16th International Conference, ASMTA 2009
Madrid, Spain, June 2009
Proceedings

Lecture Notes in Computer Science 5513

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Khalid Al-Begain Dieter Fiems
Gábor Horváth (Eds.)

Analytical and Stochastic Modeling Techniques and Applications

16th International Conference, ASMTA 2009
Madrid, Spain, June 9-12, 2009
Proceedings

Springer

Volume Editors

Khalid Al-Begain
University of Glamorgan, Faculty of Advanced Technology
Pontypridd, CF37 1DL, UK
E-mail: kbegain@glam.ac.uk

Dieter Fiems
Ghent University, Department TELIN
Sint-Pietersnieuwstraat 41, 9000 Gent, Belgium
E-mail: Dieter.Fiems@UGent.be

Gábor Horváth
Budapest University of Technology and Economics
P.O. Box 91, 1521 Budapest, Hungary
E-mail: ghorvath@hit.bme.hu

Library of Congress Control Number: Applied for

CR Subject Classification (1998): C.2, D.2.4, D.2.8, D.4, C.4

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-642-02204-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-02204-3 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12689224 06/3180 5 4 3 2 1 0

Preface

It is our great pleasure to present the proceedings of the 16th International Conference on Analytical and Stochastic Modelling Techniques and Applications (ASMTA 2009) that took place in Madrid.

The conference has become an established annual event in the agenda of the experts of analytical modelling and performance evaluation in Europe and internationally. This year the proceedings continued to be published as part of Springer's prestigious *Lecture Notes in Computer Science (LNCS)* series. This is another sign of the growing confidence in the quality standards and procedures followed in the reviewing process and the program compilation.

Following the traditions of the conference, ASMTA 2009, was honored to have a distinguished keynote speaker in the person of Kishor Trivedi. Professor Trivedi holds the Hudson Chair in the Department of Electrical and Computer Engineering at Duke University, Durham, NC, USA. He is the Duke-Site Director of an NSF Industry–University Cooperative Research Center between NC State University and Duke University for carrying out applied research in computing and communications. He has been on the Duke faculty since 1975. He is the author of a well–known text entitled *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, published by Prentice-Hall, the second edition of which has just appeared. He has also published two other books entitled *Performance and Reliability Analysis of Computer Systems*, published by Kluwer Academic Publishers, and *Queueing Networks and Markov Chains*, by John Wiley. He is also known for his work on the modelling and analysis of software aging and rejuvenation.

The conference maintained the tradition of high-quality programs with an acceptance rate of about 40%. The program of ASMTA 2009 comprisd 27 high-quality papers organized into 7 sessions. Almost every paper was peer reviewed by three reviewers from the International Program Committee. The reviewers were truly wonderful this year, as well, and in most of the cases the reviews provided valuable comments that contributed to increasing the quality of the final versions of the papers. In many cases, discussion panels were also organized when the reviews were not decisive. We would like therefore to give a special thanks to all the members of the International Program Committee for the excellent work in the reviewing process and the subsequent discussion panels during the selection process.

Keeping the tradition, ASMTA was co-located with the European Conference on Modelling and Simulation, the official conference of the European Council on Modelling and Simulation. This gave the participants of ASMTA a unique opportunity to interact with colleagues from these very relevant and complementary

areas. They also enjoyed the keynote talks delivered by prominent figures in those areas.

The local organizers made every effort to make it a memorable event. For that we give them our sincere thanks and appreciation.

June 2009

Khalid Al-Begain
Dieter Fiems
Gábor Horváth

Organization

Workshop Chair

Khalid Al-Begain	University of Glamorgan, UK

Program Chairs

Dieter Fiems	Ghent University, Belgium
Gábor Horváth	Budapest University of Technology and Economics, Hungary

Program Committee

Ahmed Al-Dubai	Napier University, UK
Sara Alouf	INRIA Sophia-Antipolis, France
Imad Antonios	Southern Connecticut State University, USA
Richard Boucherie	University of Twente, The Netherlands
Jeremy Bradley	Imperial College London, UK
Giuliano Casale	SAP Research, UK
Alexander Dudin	Belarusian State University, Belarus
Antonis Economou	University of Athens, Greece
Jean-Michel Fourneau	University of Versailles, France
Rossano Gaeta	University of Turin, Italy
Reinhard German	University Erlangen-Nuremberg, Germany
Marco Gribaudo	University of Turin, Italy
Guenter Haring	University of Vienna, Austria
Armin Heindl	University Erlangen-Nuremberg, Germany
Andras Horvath	University of Turin, Italy
Helen Karatza	Aristotle University of Thessaloniki, Greece
Remco Litjens	TNO ICT, The Netherlands
Maria Lopez-Herrero	Complutense University of Madrid, Spain
Don McNickle	University of Canterbury, New Zealand
Bruno Mueller-Clostermann	University of Duisburg-Essen, Germany
Antonio Pacheco	Instituto Superior Tecnico, Portugal
Krzysztof Pawlikowski	University of Canterbury, New Zealand
Balakrishna Prabhu	Laas-CNRS, France
Ramon Puigjaner	Universitat de les Illes Balears, Spain

Matteo Sereno	University of Turin, Italy
Bruno Sericola	IRISA/INRIA Rennes, France
Janos Sztrik	University of Debrecen, Hungary
Miklos Telek	Technical University of Budapest, Hungary
Nigel Thomas,	University of Newcastle, UK
Petr Tuma	Charles University of Prague, Czech Republic
Dietmar Tutsch	University of Wuppertal, Germany
Kurt Tutschku	University of Vienna, Germany
Benny Van Houdt	University of Antwerp, Belgium
Johan van Leeuwaarden	EURANDOM, The Netherlands
Aad Van Moorsel	Newcastle University, UK
Sabine Wittevrongel	Ghent University, Belgium
Katinka Wolter	Humboldt University of Berlin, Germany

External Referees

Tony Dale	University of Canterbury, New Zealand
Katja Gilly	Miguel Hernandez University, Spain
Uli Harder	Imperial College London, UK
Allan McInnes	University of Canterbury, New Zealand
Zsolt Saffer	Budapest University of Technology and Economics, Hungary
Hans van den Berg	University of Twente, The Netherlands
Joris Walraevens	Ghent University, Belgium

Table of Contents

Telecommunication Networks

Wireless and Mobile Networks

Simulation

Queueing Systems and Distributions

Queueing and Scheduling in Telecommunication Networks

Model Checking and Process Algebra

Performance and Reliability Analysis of Various Systems

Comparison of Multi-service Routing Strategies for IP Core Networks

Ulf Jensen and Armin Heindl

Universität Erlangen-Nürnberg
Computer Networks and Communication Systems
Erlangen, Germany

Abstract. Service differentiation in IP core networks may be supported by dedicated path selection rules. This paper investigates the degree of service distinction achievable when common routing strategies, like ECMP, SWP and WSP, are applied to two traffic classes separately and in different combinations. One traffic class requires low latencies, while the other is considered as best-effort traffic.

A Maple program has been developed that evaluates network performance characteristics, like maximal link utilization, and per-class measures, like mean end-to-end delay and mean number of hops, when paths are computed on demand with traffic demands arriving in arbitrary order. Realistic network topologies may be imported from the publicly available tool BRITE, while link capacities and traffic patterns are chosen randomly (with realistic constraints) in Maple.

Experiments show that a comparable service differentiation may already be achieved with less sophisticated strategy combinations, which apply ECMP to the delay-critical traffic class.

1 Introduction

New applications and services [1] in the next-generation Internet (NGI) require different service guarantees typically negotiated in Service Level Agreements (SLA). In order to provide appropriate service differentiation in the Internet, many proposals have been made including diverse strategies for packet classification, queue management and scheduling as well as bandwidth management and admission control. For instance, such issues are addressed in the service models for Quality of Service (QoS) support by the Internet Engineering Task Force (IETF), namely Differentiated Services (DiffServ, [2]) and Integrated Services (IntServ, [3]). Today, these technologies coexist with other approaches to provide QoS. Traffic engineering capabilities are supplied by Multi-Protocol Label Switching (MPLS, [4,5]) and QoS routing constitutes another important component in the overall QoS framework [6,7]. Architectures like GMPLS (Generalized MPLS, [8,9]) and extensions to common IP routing protocols, like OSPF (Open Shortest Path First, [10]), furnish the tools to handle traffic classes according to different rules, but it is not fully understood to which extent such routing decisions contribute to service differentiation or which rule combinations result in favorable performance.

In this paper, we investigate these issues for IP core networks. We assume that services may be set up on a semi-permanent basis, i.e., a negotiated bandwidth has to be

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 1–15, 2009.

reserved along a path or multiple paths through the network for a longer period. Service requests are served on demand, i.e., the routes cannot be optimized from a global perspective (in the knowledge of all traffic demands), but have to be computed incrementally upon arrival of a request. Such a scenario is typically encountered in traffic engineering, and its solution easily realized, e.g., by means of label-switched paths as in (G)MPLS. Furthermore, we consider two traffic classes: one traffic class is related to interactive applications and requires lower latencies, while the other represents best-effort traffic. Different routing strategies are applied to each class: we distinguish alternative link weight systems, single/multi-path routing as well as standard/QoS routing schemes, which disregard/regard the current state of the network.

A Maple program has been developed to read arbitrary topologies of realistic sizes generated with the publicly available tool BRITE [11], to assign link capacities, to generate traffic patterns for the two traffic classes and to allocate the traffic demands to the network according to specified rules. Finally, the overall network peformance is assessed by means of the maximal link utilization and the minimal unused link capacity. The service differentiation is ascertained in terms of the per-class performance measures, like the average number of hops along the paths and the mean end-to-end delays.

Just as Internet routing itself is focused on connectivity with QoS having been addressed much later, studies on traffic engineering are primarily targeted on issues like load balancing and improvement of overall network performance instead of service differentiation (e.g, [12,13,14,15,16]). While sophisticated QoS routing architectures have been proposed (e.g., [17,18]) and mostly been evaluated by discrete-event simulation on the packet level [18,19], many questions regarding fundamental design decisions remain open. In the context of Internet backbone networks, we are interested inasmuch basic routing schemes cooperate or interfere when applied to different traffic classes.

Our approach has been inspired by a case study in [15], which examines similar constraint-based routing schemes in the context of traffic engineering for a single traffic class. We extend Wang's procedure of incremental demand assignment to two traffic classes in order to study the potential of service differentiation.

The paper is organized as follows: Section 2 presents the routing schemes, which are applied in our experiments. The experiment setup and model evaluation is described in Section 3, while numerical results in Section 4 compare different routing combination for service differentiation. The paper concludes with Section 5.

2 Considered Routing Strategies

Traditional routing in the Internet is based on shortest paths between origin and destination. Each router (being the origin) computes this shortest path locally based on its view of the network. The decision to which output interface the packet is directed depends solely on the destination of the packet, which implies that all types of traffic are equally processed. In order to enable service differentiation between best-effort traffic and higher-priority traffic, current routing protocols, like OSPF [10,20], and architectures, like GMPLS [8], make provisions to manage separate routing tables for different traffic classes, which are computed based on different link metrics. Furthermore, path computation may not only be based on metric weights assigned a priori and independently of the dynamic network state, but may also reflect traffic engineering information

like the unreserved bandwidth by priority, etc. Thus, various variants of constraint-based path computation may be realized besides standard link-metric routing.

Commonly, routers periodically exchange network state information so that each one can calculate the best path(s) based on some knowledge of the entire network state conditions [17]. Other strategies have been proposed, e.g., using local information to choose a path from predefined candidate paths [21], but remote network conditions prove crucial to QoS routing. We follow the principle approach suggested in [17] that fits into the GMPLS architecture and is thus highly compatible with the existing and widely deployed routing protocols. In this paper, we focus on standard link-metric routing and QoS routing strategies solving the bandwidth-restricted path (BRP) problem [6] (via metric ordering), namely Shortest-Widest-Path (SWP) and Widest-Shortest-Path (WSP). In the context of QoS routing, we do not consider solutions to the so-called Restricted-Shortest-Path problems due to their higher complexity and neither approaches of metric combination due to their rough heuristic. Generally, QoS routing may suffer from computation and communication overhead, which hamper scalability, or from a strong snsitivity on the view of the network, where inaccuracies strongly degrade performance. These issues are only addressed in this paper in the sense that we concentrate on less sophisticated QoS routing strategies.

2.1 Standard Link-Metric Routing

Here, the link weights are set to static default values and independently of the network dynamics. In order to compute the shortest path to a destination, each router may apply classic Dijkstra algorithms [22]. Naturally, the length or cost of a path is the sum of all weights on the links between origin and destination.

Minimal Number of Hops (MH). Still today, the unit metric system is applied in large parts of the Internet. All link weights are set to 1. The shortest path minimizes the number of hops along the way between origin and destination.

Inverse Link Ratio (ILR). This metric system reflects the static link capacity. The link weight is set to the reciprocal value of the link capacity. Thus, links with high capacity attain smaller weights and are thus favored in the shortest path computation based on these weights. The rationale that traffic travels faster on high-capacity links may, however, be counteracted by attracting high loads to these links. This metric system has become very popular as a default setting in today's routers.

Equal-Cost Multi-Path (ECMP). While the above strategies are associated with single-path routing, the add-on property ECMP allows to route traffic from origin to destination on multiple shortest paths. Essentially, the ECMP rule states that – if multiple shortest paths exist – a flow is split equally. More precisely, "a flow to a destination outgoing from a node is equally split onto these outgoing links which belong to the shortest paths to this destination." (from [22], where a recursive ECMP flow allocation algorithm is given). ECMP is realized with minimal modifications of the routing tables, which now contain a next hop for every shortest path to a destination. All (or some) shortest paths can be obtained by means of the k-shortest-path-algorithm based

on any arbitrary metric, like MH, ILR, etc. ECMP balances the network load and is an important feature of OSPF with impact on traffic engineering.

2.2 QoS Routing

QoS routing finds an optimal path that satisfies a particular request under constraints, which reflect the dynamic state of the network. As a common example, the widest path maximizes the so-called bottleneck bandwidth between origin and destination [23]. The bottleneck bandwidth represents the minimal unused capacity of all links along a path. Obviously, routers must therefore exchange information on the unreserved bandwidth of links. Widest paths are well suited for load balancing, since paths with higher remaining capacities are preferred. Longer paths in terms of number of hops, however, strain the overall utilization of the network. More hops may also induce longer delays.

The Dijkstra algorithms for shortest paths are straightforwardly adapted to widest-path computations [22]. The specific changes required for the algorithms applied in this paper can be found in [24].

Since both metrics – a minimal number of hops and a high unused capacity along the path – have their benefits, paths are desired which combine these favorable properties to some extent. Solving the related bandwidth-restricted path problem implies the heuristic of metric ordering, i.e., first the best paths are found with respect to one metric and then – among these best paths – the best path with respect to the other metric is determined.

Widest Shortest Path (WSP). WSP algorithms first determine all shortest paths in terms of a standard metric (independent of network load), between which the tie is broken via the largest bottleneck bandwidth. Especially when MH is used in the first step, this metric ordering emphasizes low resource consumption in the network. WSP is computationally efficient, works well also for high network loads and/or with inaccurate network info.

Shortest Widest Path (SWP). SWP algorithms turn the metric ordering around: among all widest paths (possibly from a candidate list), the shortest one (according to a standard metric, like MH or ILR) is eventually selected. Determining widest paths first results in eventually longer paths. SWP primarily aims at load balancing. It scales well, especially in combination with path precomputation, but exhibits a more selfish behavior penalizing later requests.

In the next section, we describe in which setting these routing strategies are applied to two traffic classes in different combinations. Traffic demands of these classes incrementally routed over an initially empty network. All algorithms and computations have been implemented as Maple procedures [25].

3 Experiment Setup and Model Evaluation

Section 3.1 describes the chosen experiment setup while Section 3.2 addresses how a Maple program is used to evaluate the experiments.

3.1 Experiment Setup

An experiment comprises five steps: topology generation and read-in, traffic creation, path determination, traffic allocation and performance measure. Thereby, each traffic class uses a preassigned strategy for path determination.

Topology Generation and Read-In. At first, the tool BRITE [11] is used to generate realistic but artificial backbone topologies. In BRITE, the user can choose from a range of network models to create topologies. The employment of the Barabási-Albert model creates topologies with a majority of nodes of low degrees. The degree of a node reflects the links connected to it. Link capacities are chosen according to a discrete uniform distribution using the first seven levels of the European multiplex hierarchy. Hence, capacity values range from 51.84 Mbps to 1866.24 Mbps, which represent the lower and upper bound of the chosen multiplex levels. The links are bidirectional and share their bandwidth as needed.

Traffic Creation. The presented software produces dedicated traffic flows on traffic creation. A flow from source s to destination t will be distinguished from a flow from source t to destination s. Prior to traffic demand creation, the nodes are split in boundary nodes and transit nodes. Traffic demands use boundary nodes as source and destination, transit nodes are only used as intermediate nodes. The partitioning into boundary and transit nodes is carried out considering the degree of every node and the average node degree. For details of this procedure see [24]. Once the nodes are partitioned, static traffic demands can be established. A traffic volume drawn from a uniform distribution in an arbitrary, but fixed interval is assigned to each demand. The traffic generation process is the same, no matter how many traffic classes are used. To conclude this step it can be stated that after splitting in transit and boundary nodes, the desired amount of dedicated demands are established between unique pairs of boundary nodes.

Path Determination and Traffic Allocation. Traffic classes may assume different shares of the overall traffic. This means that each class has a fraction of demands from the set of overall demands. For example, if the best-effort class and the delay-critical class have the same portion of traffic and ten traffic demands are to be allocated, each class has to realize five demands. Starting with the total amount of estimated traffic demands, each demand is assigned to a traffic class as follows.

1. A demand is randomly chosen.
2. The traffic class that will realize this demand is chosen. If both classes still require demands, one class is randomly picked. Otherwise, the demand will be realized by the only class which still has traffic to carry. This procedure ensures a ”mixing” during the allocation of demands of the different traffic classes.
3. Having chosen a traffic class the demand belongs to, the routing path needs to be determined using the strategies described in Section 2. In case of multi-path routing (e.g. ECMP), potentially more than one path has to be identified.
4. The allocation of traffic simply means substracting the demand volume from the capacities of all links along the routing path(s).

With this procedure, the network capacity will be downsized step by step with every allocated demand.

Performance Measures. In the final experiment step, performance measures are computed. We first discuss network measures, which assess the aggregate performance of the routing, and then per-class measures, which manifest the service differentiation between the traffic classes. Having saved the idle topology (i.e., with the original capacities fully available), it is easy to determine network characteristics using the loaded and idle network. The maximal link utilization and the minimal unused link capacity are determined. The formula

$$U = \max\left\{\frac{y_e}{c_e} : e \in E\right\}$$

purveys the maximal link utilization in percent, where E denotes the set of edges/links and c_e their capacity. The term

$$y_e = \sum_d \sum_{p=1}^{P_d} \delta_{edp} x_{dp}$$

describes the traffic on link e. The variable x_{dp} contains the demand volume that is routed on path p of the P_d paths which realize demand d. The logical value δ_{edp} determines whether path p (realizing demand d) uses link e ($\delta_{edp} = 1$) or not ($\delta_{edp} = 0$). The minimal unused link capacity is computed in Mbps by using the formula

$$C = min\left\{c_e - y_e : e \in E\right\} \quad .$$

To calculate traffic characteristics for each traffic class, the routing paths for each traffic demand are needed. This data, together with the idle and loaded network, is the base to compute the average number of hops on the path and the mean path delay. The mean path delay depends on the link delay of each link on the path. The link delay is approximated with the formula

$$D_e = \frac{1}{c_e - y_e} \quad ,$$

in analogy to the time in the system of an M/M/1-queue. This rather rough approximation is sufficient for our purposes of a relative comparison. The mean path delay, weighted with the demand volume, is computed using the formula

$$D = \sum_d \sum_{p=1}^{P_d} \frac{x_{dp}}{\sum_d h_d} \sum_e \delta_{edp} \frac{1}{c_e - y_e} \quad . \tag{1}$$

The variable h_d represents the demand volume of demand d. This delay includes transmission and queueing delay and is denoted in seconds. Another important traffic characteristic is the average number of hops which is also needed to compute the processing delay. The formula

$$H = \sum_d \sum_{p=1}^{P_d} \frac{x_{dp}}{\sum_d h_d} \sum_e \delta_{edp}$$

is used to compute this characteristic. The paths are again weighted according to their demand size. To estimate the end-to-end delay of the traffic of a traffic class, presumptions about the propagation delay and the processing delay per hop have to be made.

This means, that both traffic characteristics shall be contemplated together with network presumptions for the propagation and processing delay.

3.2 Usage of Maple Software

Executing the above-mentioned five steps results in network and traffic characteristics for a fixed amount of demands and a specified demand ratio of the traffic classes. To extensively evaluate the performance of two routing strategies, we obtain values for different amounts of demands. The use of different demand ratios is of high practical interest. In addition, the randomness in traffic generation and demand assignment in an experiment is coped with by executing an adequate amount of repetitions. The developed Maple program provides for independent replications. Comprehensive experiments are executed with a single function call, while the architecture allows simple enhancement and adaption. The following section deals with usage, prospects and adaption options of the software.

An experiment to compare two routing strategies that are assigned to a traffic class each consists of different partial experiments. In these partial experiments, the portion of demands of each traffic class on the overall demand amount differs. The results of Section 4 use the ratios presented in Table 1. The ratios can easily be adapted to satisfy practical situations.

Table 1. Share of demands of each traffic class of the overall demand amount

	partial experiment			
traffic class	1	2	3	4
best effort	100%	90 %	70%	50%
delay critical	0%	10 %	30%	50%

For the traffic classes, every desired combination of routing strategies can be chosen. The strategies described in Section 2 are just a selection of common policies. Other strategies can be easily added and are specified as an argument when calling the experiment function which executes automatically the partial experiments. Another argument is the ratio of boundary and transit nodes. Sufficiently many boundary nodes are needed to create more unique source-destination-pairs as traffic demands are specified. The interval for the demand volumes is an argument, too. The number of repetitions may be specified as a tradeoff between statistical significance and execution time.

The developed program offers the opportunity of executing sophisticated experiments that deliver extensive results while keeping usage easy. Numerous arguments allow simple adaption and the modular architecture ensures easy enhancement.

4 Comparison of Multi-service Routing Schemes

This section presents results of experiments conducted with a topology of 40 nodes and 77 links (as generated with BRITE). The link capacities are uniformly chosen from the first seven layers of the European multiplex hierarchy. Further preferences are an

interval between 3 and 12 for the demand volumes and the portion of 0.9 of all nodes for the boundary nodes. The experiment starts with 20 demands and increases its number by 10 up to 130. For the latter traffic, a maximum link utilization of around 60% is attained for every experiment, which is considered as a reasonable operational load.

The result figures for the network performance characteristics show one curve for each partial experiment (see Table 1). The solid curve displays the behavior without traffic differentiation. In this case, the whole traffic is routed using the best-effort strategy. The dashed line represents a share of 10% delay-critical traffic of the overall demands. Partial experiment 3, illustrated in the dashed-dotted line, has a share of 30% delay-critical traffic while the dotted line shows the results of the experiment where both traffic classes realize the same number of demands.

One partial experiment is chosen to display the per-class traffic characteristics. The figures show partial experiment 3 with a share of 30% of the overall demands for the delay-critical traffic class. For comparison, the traffic characteristics of partial experiment 1 with no traffic differentiation are shown as well.

To produce the results of each subsection (with seven replications in each specific setting, i.e., for each demand number in each partial experiment), run times on standard PCs varied between one hour for the simpler strategies and some hours for more complex experiments with QoS routing. We do not show confidence intervals for our results in order not to overload the figures. They can be found in [24].

Among the various strategies we have evaluated, we show here three routing combinations which highlight the potential of service differentiation without and with QoS routing strategies.

4.1 ECMP (ILR) for Best Effort and ECMP (MH) for Delay-Critical Traffic

This combination was selected to find out whether strategies that do not use QoS routing techniques are able to provide service differentiation by different weight assignments. ECMP with the MH strategy is selected for the delay-critical traffic, since shortest paths with the minimal number of hops minimize the per-node delays, while link loads are somewhat balanced by ECMP leading to lower uilizations and thus lower delays. Note that ECMP splits a demand over different paths of the same length. Best-effort traffic is merely engineered to balance the network load. Besides the ECMP multi-path routing, the ILR link weight assignment tends to direct traffic over possibly longer paths with links of higher capacity. The static weights of inverse link capacities allow to avoid bottlenecks slightly better than unit weights.

Figure 1 displays the network performance characteristics. According to intuition, as the load on the network increases, the maximal link utilization increases and the minimal unused link capacity decreases. With respect to the different partial experiments, we only discuss the maximal link utilization. In partial experiment 1 (solid line), all traffic is routed according to ECMP (ILR), i.e., without any traffic differentiation. Compared with the other partial experiments with two traffic classes, this weight assignment due to link capacities yields benefits in low network loads (see lower maximal link utilization), but tends to show worse network performance for higher loads. In our experiment setting, the maximal link utilization reaches values of beyond 80 %. Also with increasing load, ILR prefers high-capacity links, which become heavily loaded. This effect is

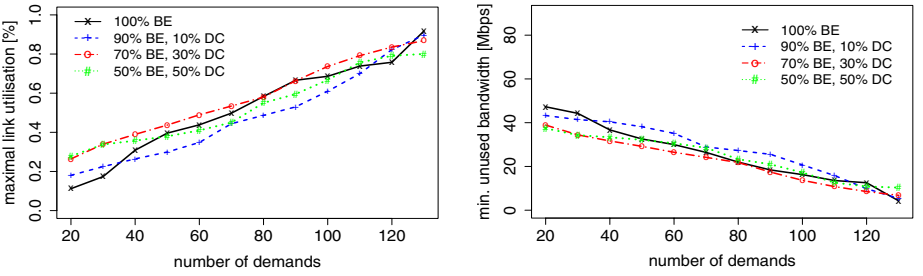


Fig. 1. Network characteristics for experiment with ECMP (ILR) for best-effort class and ECMP (MH) for delay-critical class: maximal link utilization is shown left, minimal unused link capacity on the right for different partial experiments

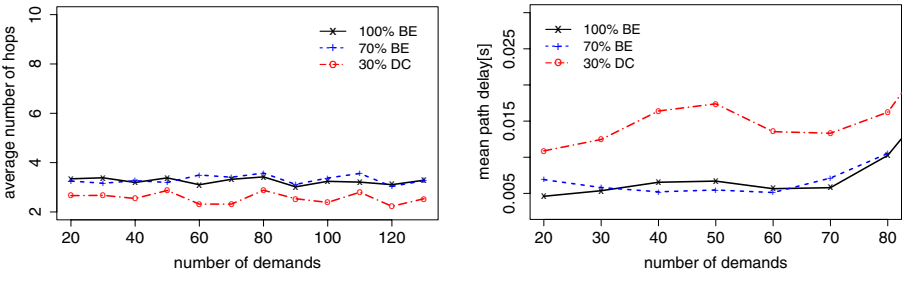


Fig. 2. Traffic characteristics for experiment with ECMP (ILR) for best-effort class and ECMP (MH) for delay-critical class: average hop count is shown left, mean path delay on the right for partial experiments 1 and 3

mitigated in the presence of delay-critical traffic, which is routed according to different rules, which explains the relatively bad performance of partial experiment 1.

In the center range, best network performance is achieved for the partial experiment 2 with a 10 % share of delay-critical traffic (see dashed curve). With higher such shares, network performance deteriorates again, even more for 30 % (dashed-dotted line) than for 50 % (dotted line). Obviously, small shares (around 10%) of delay-critical traffic may have a positive influence on the network performance. We assume that the load on lower-capacity links is too small to influence the overall network performance in partial experiment 2.

The traffic characteristics per class are shown in Figure 2 for partial experiment 3 (30 % share of delay-critical traffic) along with reference curves for partial experiment 1. Considering the average hop count (left), the curves reveal that the delay-critical class traffic (dashed-dotted line) is routed on shorter paths as desired to reduce the per-node overhead. The solid line of partial experiment 1 (100 % best-effort traffic) and the dashed line of the best-effort share are quite similar, with the dashed line assuming slightly larger values in most cases. A difference of one hop (on average) can be observed between best-effort and delay-critical traffic and substantiates the presumptions for choosing this strategy combination. The MH weight assignment accounts for shorter routing paths, while ILR pays the price for avoiding low-capacity links with

longer paths. The paths of the best-effort class are even longer than the paths of the partial experiment 1 without traffic differentiation, because bottleneck links caused by the delay-critical class need to be avoided additionally. A small but considerable differentiation can be stated for the traffic characteristic of the average hop count.

The curves for the mean path delay are shown up to an amount of 80 demands. Above this number, meaningful results could not be obtained due to overload situations. With the maximal link utilization (averaged over 7 replications) beyond 60 %, the probability that a single link in one of the replications becomes overloaded increases. In such a case, the mean path delay (see equation (1)) can no longer be computed [24].

The curve constellation in Figure 2 (right-hand side) showing the mean path delay is counterintuitive at first sight. The dashed-dotted line of the delay-critical class ranks above the best-effort curve meaning that a best-effort traffic is routed on paths with lower mean delays. Since the delay-critical traffic uses fewer hops on average, it must traverse links which are more heavily loaded. According to (1), the link delays are the crucial factor. Nevertheless, another issue needs to be adressed. The path delay formula considers only transmission and queueing delays and does not regard propagation and processing delays. For mean end-to-end delays, propagation and processing delays have to be considered. Then, both traffic characteristics have to be evaluated together, because processing delays are added for every hop and propagation delays for every link used on the routing path. An assumption of 10 to 15 ms per hop for processing and propagation delay is reasonable for the considered IP core networks and leads to a different situation for the evaluation of the end-to-end delay for the different traffic classes. A difference of 10 ms on average for the mean path delay and one hop on average for the hop count (as roughly shown by Figure 2) leads to the conclusion that the traffic will only be differentiated with respect to the mean end-to-end delays, if the processing and propagation delays rise above 10 ms. That means, to make clear predictions, details about the network topology, like distances between nodes and node behavior, need to be known. The strategy combination considered in this subsection is only useful for traffic differentiation (for the mean end-to-end delays), if processing and propagation delays are considerably larger than 10 ms.

4.2 SWP (MH) for Best-Effort and ECMP (ILR) for Delay-Critical Traffic

This experiment applies the QoS routing strategy SWP to the best-effort class to further exploit load balancing for this traffic class. The MH weight assignment in SWP finds the shortest path (in terms of number of hops) of a set of widest paths. Since widest paths take into account the dynamic state of the network, load balancing is improved. The predominant IP routing strategy ECMP with weight determination ILR is used for the delay-critical class and is based on static network properties.

As before, Figure 3 displays the network characteristics and shows similar reciprocal trends for both network performance characteristics. However, with respect to the previous experiment, the network performance is now considerably improved: for 130 demands, the maximal link utilization is now around 60 % (as opposed to 80 %) and the minimal unused link capacity remains over 20 Mbps (as opposed to less than 10 Mbps). As a consequence, all curves show smaller slopes than in the previous experiment.

The QoS routing strategy for the best-effort class made load balancing more effective and stable. All curves in each figure are quite close to each other, where the partial experiments with a higher share of delay-critical traffic (dotted and dashed-dotted lines) show slightly worse network performance. For higher loads, the ordering of the partial experiments is as expected: with an increasing share of the delay-critical traffic (with non QoS routing in this experiment), the network performance deteriorates. The good performance of partial experiment 1 (10 % delay-critical traffic, dashed line) in the center part may be attributed to the suitable ILR weight assignment for low network load.

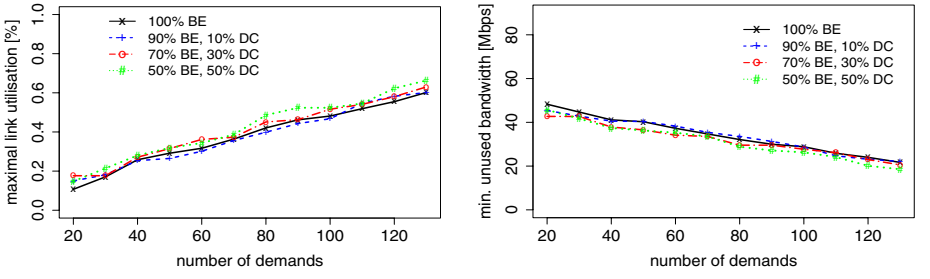


Fig. 3. Network characteristics for experiment with SWP (MH) for best-effort class and ECMP (ILR) for delay-critical class: maximal link utilization is shown left, minimal unused link capacity on the right for different partial experiments

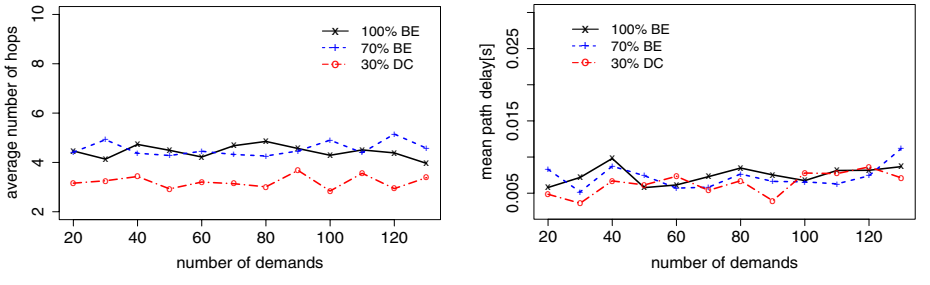


Fig. 4. Traffic characteristics for experiment with SWP (MH) for best-effort class and ECMP (ILR) for delay-critical class: average hop count is shown left, mean path delay on the right for partial experiments 1 and 3

Figure 4 shows the per-class traffic characteristics. A significant differentiation can be observed. With respect to the mean number of hops (see left-hand side), the quantitative difference, estimated to about one and a half hops, is larger than the one in the previous experiment. The values for ECMP (ILR) remained rather unchanged, while SWP (MH) uses paths that contain more hops on average. This use of widest paths results in a considerable differentiation of the traffic classes for this characteristic.

The second traffic characteristic on the right side of Figure 4 appears quite different compared with the experiment of Section 4.1. Due to the well-balanced network load, meaningful results could be achieved for demand numbers up to 130. In addition, the curve constellation itself is remarkable as both traffic classes reach quite similar values.

The dashed-dotted line for the delay-critical traffic lies below the dotted line for the best-effort traffic for most of the calculated values. As intended, the best-effort class which uses longer paths on less loaded links now appears to encounter longer delays than the delay-critical class, with uses potentially shorter paths with high capacities. Considering both traffic characteristics together and assuming the same 10 ms for processing and propagation delay as in the previous experiment, we conclude that a significant service differentiation can be reached with respect to mean end-to-end delays with the strategy combination chosen for this experiment. The delay-critical class traffic will be routed faster even without taking topology details into account. The performance benefit will further increase with higher values for the processing and propagation delay.

Further experiments with the same strategy combination were accomplished to back up the observed results. On the one hand, the topology size was raised while using the same network generation model. On the other hand, another generation model, the Waxman model [11], was utilized in BRITE while keeping the network size constant. Other preferences as described in Section 3.1 were not changed.

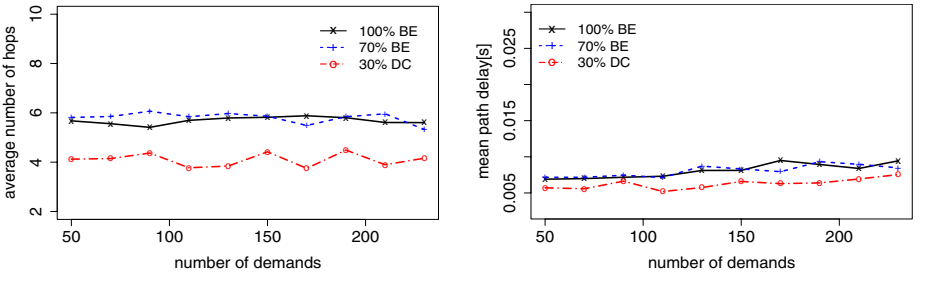


Fig. 5. Traffic characteristics for 50 node experiment with SWP (MH) for best-effort class and ECMP (ILR) for delay-critical class: average hop count is shown left, mean path delay on the right for partial experiments 1 and 3

Figure 5 shows the traffic characteristics for the experiment with a 50 node topology and same network generation model usage. The results for the application of the Waxman model show the same trends and can be extracted from [24]. The figures reveal, that the number of demands was increased notedly. In a larger network, more traffic is needed to reach a adequate operation load. With the higher amount of demands, the curves are mainly flattened as the randomness in traffic generation and demand assignment is balanced.

The average hop count curves that appear on the left side show a significant differentiation with a difference of two hops in average. The values for all curves increased with the network size as there are simply longer paths needed on the way from source to destination. The best-effort class (dashed line) uses six hops while the delay-critical class needs four hops on the path. The increased difference in comparison to the 40 node network experiment can also be imputed to the larger topology.

Contemplating the right figure with the mean path delay, it can be stated that the dashed-dotted line (delay-critical class) lies below the dashed line of the best-effort class. SWP, as used for the best-effort class, uses more hops and adds path delay for

each link utilized. Although these links are few loaded, WSP that uses less but potentially higher loaded links finds paths with less queuing and transmission delay in average. This differentiation, that appeared already as a trend in the 40 node experiment, is significant and therefore a good capability to differentiate traffic can be attested this strategy combination.

The comparing experiment with a 50 node network revealed, that this strategy combination is able to differentiate the traffic classes in both traffic characteristics. A closer examination of the topology is no longer needed. The trends assumed for the curves shown above were approved.

4.3 SWP (MH) for Best Effort and WSP (ILR) for Delay-Critical Traffic

This section presents another promising strategy combination in our experiments. Now, QoS routing strategies are applied to both traffic classes, namely SWP (MH) for best effort (as before) and WSP for the delay-critical class (as opposed to ECMP). WSP with ILR weight assignment was chosen to encourage the use of short paths while considering the bottleneck bandwith as second routing metric. In combination with SWP, load balancing is expected to be further improved for this experiment. Both strategies reach their routing decision based on the dynamic state of the network, while SWP focuses on using the widest path and WSP on minimizing the hop count.

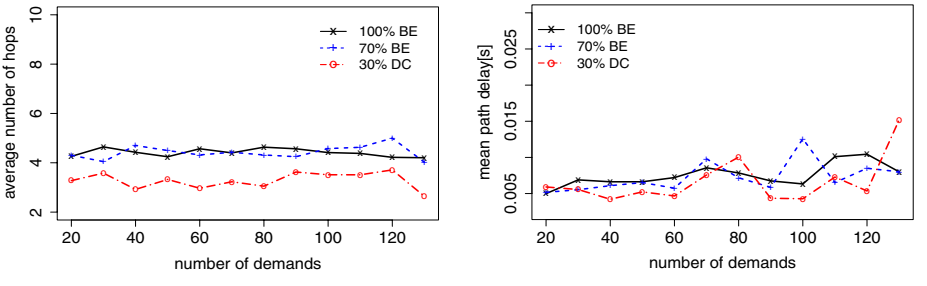


Fig. 6. Traffic characteristics for experiment with SWP (MH) for best-effort class and WSP (ILR) for delay-critical class: average hop count is shown left, mean path delay on the right for partial experiments 1 and 3

The results for the network performance characteristics look alike to the previous 40-node experiment with slightly better values and are not shown here. For figures and further details see [24].

Figure 6 shows the traffic characteristics for this experiment. In comparison to the previous experiment, the curves that display the average hop count are quite similar. On a closer look, they appear to be slightly closer together what may be due to a better load balancing capability of WSP. A quantitative difference of one to one and a half hops can be observed what leads again to a significant differentiation for this characteristic.

With respect to the curves displaying the mean path delays, it can be observed that the dashed and dashed-dotted line are close to each other. Again, the delay-critical class seems to reach somewhat better values thus faster paths are suggested. This strategy combination does not reach a clear differentiation in this traffic characteristic in this

scenario but can be identified as suited for an end-to-end delay traffic differentiation. Therefore, the average hop count and assumptions concerning the processing and propagation delays need to be considered together as in the previous experiment evaluation.

The strategy combinations SWP (MH)/WSP (ILR) and SWP (MH)/ECMP (ILR) are quite comparable and their results reveal a good ability to differentiate traffic in IP core networks. Unexpectedly, no major improvements are detected for the application of QoS routing strategies at both traffic classes. For practical reasons ECMP (ILR) should be preferred over WSP (ILR), as the determination of the bottleneck bandwidth is computationally more complex than the determination of shortest paths with static routing weights.

5 Conclusions

The next-generation Internet may tune various different QoS mechanisms to achieve service differentiation between traffic classes. This paper investigates for IP backbone networks with on-demand routing to which extent per-class routing may contribute to this goal. A rather flexible computational framework has been developed in Maple to quantitatively assess various combinations of standard link-metric routing and/or QoS routing strategies.

Results for two traffic classes – delay-critical and best-effort – have shown that a noticeable service differentiation in terms of mean end-to-end delays and mean number of hops may already be achieved with rather fundamental routing schemes. Best results were obtained when SWP (with shortest paths according to MH) is used for the best-effort traffic, while the delay-critical traffic is routed according to WSP (ILR) or even ECMP (ILR). The comparable performance in these two cases is remarkable, since ECMP (ILR) does not take into account the dynamic state of the network. In any case where non-QoS routing is applied, typically for the best-effort class, the ECMP feature is crucial in order to achieve some service differentiation.

In future work, more complex QoS routing approaches will be considered in order to assess the additional benefit of more sophisticated routing procedures – also in the context of more distinct traffic classes with other QoS requirements.

References

1. Toguyeni, A., Korbaa, O.: Quality of service of Internet service provider networks: State of the art and new trends. In: Proc. of Int. Conference on Transparent Optical Networks (2007)
2. Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An architecture for differentiated service. IETF RFC 2475 (December 1998)
3. Braden, R., Clark, D., Shenker, S.: Integrated Services in the Internet Architecture: An Overview. IETF RFC 1633 (June 1994)
4. LeFaucheur, F., et al.: Multi-Protocol Label Switching (MPLS) Support of Differentiated Services. IETF RFC 3270 (May 2002)
5. Colitti, W., Steenhaut, K., Nowe, A.: Multi-layer traffic engineering and DiffServ in the next generation Internet. In: Proc. of 3rd Int. Conference on Communication System Software and Middleware (COMSWARE 2007), pp. 591–598 (2007)

6. Curado, M., Monteiro, E.: A survey of QoS routing algorithms. Trans. Engineering, Computing and Technology (2004)
7. Alsharif, S., Shahsavari, M.M.: Performance study of MPLS and DS techniques to improve QoS routing for critical applications on IP networks. In: Proc. of SPIE - The International Society for Optical Engineering (2008)
8. Bryskin, I., Farrel, A.: GMPLS: Architecture and Applications. Morgan Kaufmann Publishers Inc., San Francisco (2005)
9. Moungla, H., Krief, F.: Service differentiation over GMPLS. In: de Souza, J.N., Dini, P., Lorenz, P. (eds.) ICT 2004. LNCS, vol. 3124, pp. 628–637. Springer, Heidelberg (2004)
10. Guerin, R., Orda, A., Williams, D.: QoS routing mechanisms and OSPF extensions. In: Proc. of Global Internet Miniconference (1997)
11. Medina, A., Lakhina, A., Matta, I., Byers, J.W.: BRITE: An approach to universal topology generation. In: Proc. of 9th Int. Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2001), p. 346. IEEE Computer Society, Los Alamitos (2001)
12. Leduc, G., Abrahamsson, H., Balon, S., Bessler, S., D'Arienzo, M., Delcourt, O., Domingo-Pascual, J., Cerav-Erbas, S., Gojmerac, I., Masip-Bruin, X., Pescapé, A., Quoitin, B., Romano, S.F., Salvatori, E., Skivée, F., Tran, H.T., Uhlig, S., Umit, H.: An open source traffic engineering toolbox. Computer Communications 29, 593–610 (2006)
13. Fortz, B., Thorup, M.: Internet traffic engineering by optimizing OSPF weights. In: Proc. of 19th IEEE INFOCOM, Tel Aviv, Israel, vol. 2, pp. 519–528 (2000)
14. Fortz, B., Rexford, J., Thorup, M.: Traffic engineering with traditional IP routing protocols. IEEE Communications Magazine 40, 118–124 (2002)
15. Wang, Z.: Internet QoS: Architectures and Mechanisms for Quality of Service. Morgan Kaufmann Publishers Inc., San Francisco (2001)
16. Raja, S.V.K., Raj, P.H.: Integrated subset split for balancing network utilization and quality of routing. Proc. of World Academy of Science, Engineering and Technology 20, 96–100 (2007)
17. Crawley, E., Nair, R., Tajagopalan, B., Sandick, H.: A framework for QoS based routing. IETF RFC 2386 (August 1998)
18. Varela, A., Vazao, T., Arroz, G.: Multi-service routing: A routing proposal for the next generation Internet. In: Boavida, F., Plagemann, T., Stiller, B., Westphal, C., Monteiro, E. (eds.) NETWORKING 2006. LNCS, vol. 3976, pp. 990–1001. Springer, Heidelberg (2006)
19. Ash, G.: Traffic Engineering and QoS Optimization of Integrated Voice & Data Networks. Morgan Kaufmann Publishers Inc., San Francisco (2007)
20. Kompella, K., Rekhter, Y.: OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS). IETF RFC 4203 (October 2005)
21. Nclakuditi, S., Zhang, Z., David, H.C.D.: On selection of candidate paths for proportional routing. Computer Networks 44, 79–102 (2004)
22. Medhi, D., Ramasamy, K.: Network Routing: Algorithms, Protocols, and Architectures. Morgan Kaufmann Publishers Inc., San Francisco (2007)
23. Wang, Z., Crowcroft, J.: Quality-of-service routing for supporting multimedia applications. IEEE Journal on Selected Areas in Communications 14(7), 1228–1234 (1996)
24. Jensen, U.: Vergleich von Traffic Engineering Strategien fuer IP-Netze mit verschiedenen Verkehrsklassen. Diplomarbeit, Fach Informatik, University of Erlangen, Germany (2008)
25. Maplesoft: Maple 12. Tool for mathematics and modeling, Waterloo, Ontario, Canada (2009), http://www.maplesoft.com/

Analysis of Opportunistic Spectrum Access in Cognitive Ad Hoc Networks

Mohamed A. Kalil[1], Hassan Al-Mahdi[2], and Andreas Mitschele-Thiel[1]

[1] Faculty of Computer Science and Automation,
Ilmenau University of Technology, Germany
{mohamed.abdrabou,mitsch}@tu-ilmenau.de
[2] Faculty of Computers and Informatics,
Suez Canal University, Egypt
hassanwesf@yahoo.com

Abstract. Cognitive radio (CR) is a promising technology for increasing the spectrum capacity for ad hoc networks. Based on CR, the unlicensed users will utilize the unused spectrum of the licensed users in an opportunistic manner. Therefore, the average spectrum usage will be increased. However, the sudden appearance of the licensed users forces the unlicensed user to vacate its operating channel and handoff to another free one. Spectrum handoff is one of the main challenges in cognitive ad hoc networks. In this paper, we aim to reduce the effect of consecutive spectrum handoff for cognitive ad hoc users. To achieve that, the licensed channels will be used as operating channels and the unlicensed channels will be used as backup channels when the primary user appears. Therefore, the number of spectrum handoff will be reduced, since unlicensed bands are primary user free bands. A Markov chain model is presented to evaluate the proposed scheme. Performance metrics such as blocking probability and dropping probabilities are obtained. The results show that the proposed scheme reduces all the aforementioned performance metrics.

Keywords: Cognitive radio, Markov chain, ad hoc networks.

1 Introduction

A significant portion of the spectrum in the licensed band (e.g. TV band) is not utilized [1]. On the contrary, ad hoc networks nowadays are managed by a static spectrum allocation which leads to spectrum inefficiency problem. To overcome this problem, the concept of cognitive radio (CR) [2], [3] was introduced. In CR networks, there are two types of users: the licensed or primary users (PUs) and the unlicensed or secondary users (SUs). The SUs can periodically search for and determine unused channels in the licensed band. Based on the scan results, SUs can communicate with each other without interfering the PUs. In [4], [5], an interesting and brief overview of CR and current challenges in this technology are introduced. One of the main challenges, that affects the performance of the SU, is the sudden and consecutive appearance of a PU. In such a case, an SU

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 16–28, 2009.

is forced to vacate the occupied channel to another free channel. This process continues until the SU finishes its transmission. This is called spectrum handoff process. This process leads to a high transmission delay for SUs. Therefore, spectrum handoff should be reduced. According to [4] and [6], SUs on neXt Generation (XG) networks can operate in both licensed and unlicensed bands. However, to the best of our knowledge, most of the researchers are focusing on the behavior of SUs in the licensed band, supposing that the unlicensed band is already saturated and therefore the effect of unlicensed bands is neglected. The possibility that the unlicensed band may become free is not taken into their consideration.

In this paper, a new scheme for spectrum access in a heterogeneous spectrum environment of licensed and unlicensed bands is introduced. We believe that most of the wireless devices in the future will have CR capabilities and only few devices will be wireless devices without CR support. Since the licensed bands cover a large geographical area and a significant portion of this spectrum is unused, we suppose that an SU will utilize the licensed channels as operating channels and the unlicensed channel as backup channels in case of the appearance of PUs. The advantages of using the unlicensed channels as backup channels are twofold: 1) the dropping probability will be reduced in case of the appearance of PUs. 2) the number of spectrum handoff is reduced. A general Markov chain model, to investigate the performance of SUs in this heterogeneous spectrum environment, is presented. Based on this model, different performance metrics such as blocking probability, dropping probability and throughput are derived. This model is compared with the classical opportunistic spectrum access model. As a result of using the unlicensed channels as backup channels, the blocking and dropping probabilities for SUs are decreased. Furthermore, the throughput is increased.

The rest of this paper is organized as follows: An overview about related work is presented in section 2. The analytical models for the classical opportunistic spectrum access scheme and the proposed scheme are introduced in section 3. In section 4, the numerical results are illustrated. Finally, summary and conclusion are presented in section 5.

2 Related Work

A number of analytical models for opportunistic spectrum access (OSA) schemes have been introduced recently in the literature. However, these analytical models evaluate the performance of SUs in the licensed band or the unlicensed bands separately. The authors do not take into their considerations that SUs may operate over both bands. They ignore that the unlicensed band may become free after some time and therefore may be used again. Therefore, the effect of unlicensed channels on the behavior of SUs was neglected. In their models, the SUs are accessing the unused spectrum of PUs opportunistically. In case of an appearance of a PU, the transmission of the SU will be stopped until another free primary channel becomes free, otherwise the SU will be dropped.

In [7], a Markov chain analysis for spectrum access in licensed bands for cognitive radios was presented. The author derived the blocking probability and dropping probability for SUs operating in the licensed band only. In [8], a Markov chain model has been introduced to predict the behavior of open spectrum access in unlicensed bands only.

In [9], the performance of SUs in spectrum sharing with PUs has been evaluated through a three-dimension Markov chain model. However, SUs spectrum handoff is not presented in their model.

In [10], an efficient and fair MAC protocol as well as QoS provisioning for a CR device, while coexisting with the legacy users on both licensed and unlicensed bands, has been proposed. However, an analytical model was not investigated in their work.

In [11], the spectrum usage for SUs is increased by setting the licensed channels as the operating channels, because it covers a large geographical area and a significant portion of this spectrum is unused. Furthermore, the unlicensed channels will be used as backup channels in case of the appearance of PUs. However, a detailed analytical model was not presented in this work.

The main contribution of this paper is to extend the work, done in [11], by evaluating analytically the performance of SUs in a heterogeneous spectrum environment of licensed and unlicensed bands. A general Markov chain model is presented and compared with the classical opportunistic spectrum access model.

3 Analytical Model

In this section, the analytical models for the classical opportunistic spectrum access (OSA) and the proposed scheme named opportunistic spectrum access with backup channels (OSAB) will be analyzed. The common assumptions for both schemes are summarized in the following section.

3.1 Common Assumptions

In this section, the common assumptions will be presented as follows.

- There are two types of available spectrum (channels), licensed and unlicensed channels. The licensed channels are named primary channels, while the unlicensed channels are named secondary channels.
- The maximum numbers of primary and secondary channels within the transmission range of a given node are assumed to be c_1 and c_2, respectively.
- The total number of available channels for SUs depends on the number of busy primary channels. This number is given as $g_i = c_1 + c_2 - i$, where i is the number of busy primary channels.
- The arrival process of PUs and SUs is assumed to be Poisson with rate λ_1 and λ_2, respectively.
- The service times of the PUs and SUs are assumed to be an exponential distribution with expectation $\frac{1}{\mu_1}$ and $\frac{1}{\mu_2}$ respectively.

For simplicity, it is assumed that the nodes of the cognitive ad hoc network under consideration are all homogeneous, i.e. statistically identical and independent.

3.2 Opportunistic Spectrum Access (OSA)

In this section, the classical OSA model is evaluated. Based on this model, the SUs operate in the unoccupied primary channels. If the PU appears, an SU shall immediately handoff from the current channel to another free one. If there are no free primary channels, an SU is dropped. An SU is dropped, even if there are free secondary channels. Therefore, the secondary channels have no effect on the performance of SUs. The process of spectrum access to the primary channels is modeled as a two-dimensional Markov chain. The number of primary channels, c_1 will be sharable between PUs and SUs. Therefore, states in the transition diagram are described by (i, j), where i is the number of primary channels used by PUs and j is the number of primary channels used by the SUs. The state space S is given by

$$S = \{(i, j) \mid 0 \leq i \leq c_1, 0 \leq j \leq c_1 - i\}$$

Let $p_{i,j}$ be the steady-state probability distribution for a valid state $(i, j) \in S$. The state transition diagram for this scheme is given by Figure 1.

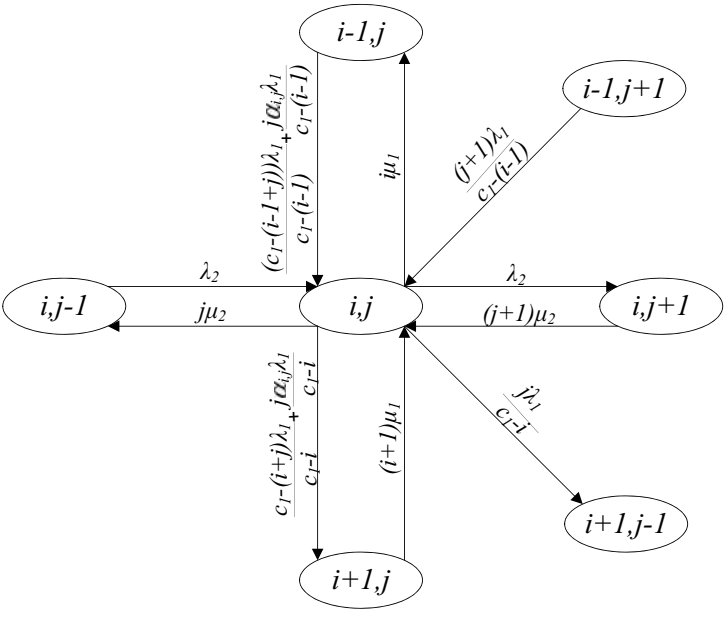


Fig. 1. Markov chain model for spectrum sharing without backup channels

The state (i, j) can be moved to one of the following states depending on the arrival of PUs or SUs.

1. State $(i+1, j)$. This case can be reached, if either one of the two following events happens:
   - a PU arrives and occupies a free channel which is not utilized by an SU with probability $\frac{c_1-(i+j)}{c_1-i}$. Therefore, the ongoing SU transmission will not be affected.
   - a PU arrives and occupies a channel which is utilized by an SU with probability $\frac{j\alpha_{i,j}}{c_1-i}$, whereas the indicator variable $\alpha_{i,j} = 1$, if $i + j < c_1$ and 0 otherwise. Therefore, the PU preempts the SU from this channel. Furthermore, the preempted SU performs a handoff to another free one $i + j < c_1$.
2. State $(i+1, j-1)$. This case can be reached if a PU arrives and operates in the same channel that is occupied by an SU with probability $\frac{j}{c_1-i}$. Furthermore, there are no primary channels available for the SU to complete its transmission. Therefore, the SU will be preempted and dropped.
3. State $(i, j+1)$. This case can be reached if an SU arrives and operates in a free primary channel.

Furthermore, the service completion for both the PU and the SU moves state (i, j) to states $(i-1, j)$ and $(i, j-1)$, respectively. Based on the state transition diagram in Figure 1, the steady-state balance equation for $p_{i,j}$ is given as follows:

For $0 \leq i \leq c_1$ and $0 \leq j \leq c_1 - i$.

$$\begin{aligned} A_{i,j}p_{i,j} = &\left(\frac{(c_1-(i-1+j))}{c_1-(i-1)} + \frac{j\alpha_{i,j}}{c_1-(i-1)}\right)\lambda_1 p_{i-1,j} \\ &+\frac{(j+1)}{c_1-(i-1)}\lambda_1 p_{i-1,j+1} + \lambda_2 p_{i,j-1} \\ &+(i+1)\mu_1 p_{i+1,j} + (j+1)\mu_2 p_{i,j+1} \end{aligned} \tag{1}$$

where $p_{i,j} = 0$ for $i < 0$, $j < 0$ or $i + j > c_1$. The value of $A_{i,j}$ is given as

$$A_{i,j} = \frac{(c_1 - i + j\alpha_{i,j})}{c_1 - i}\lambda_1 + \lambda_2 + j\mu_2 + i\mu_1$$

An iterative technique will be adopted to obtain the steady-state probabilities $p_{i,j}$. Once these probabilities are obtained, some performance metrics can be calculated. Note that, the following algorithm converges as long as $\lambda_1 + \lambda_2 < c_1(\mu_1 + \mu_2)$.

1. Set a certain convergence threshold κ.
2. Input: c_1, λ_1, λ_2, μ_1 and μ_2.
3. Initialize $p_{i,j}^{old} = 1$ for $i = j = 0$ and $p_{i,j} = 0$ for $i + j > 0$.
4. Compute the probabilities $p_{i,j}^{new}$ using (1).
5. If $\left|p_{i,j}^{new} - p_{i,j}^{old}\right| > \kappa$, then set $p_{i,j}^{old} = p_{i,j}^{new}$ and go to Step 4.
6. Once the steady-state probabilities $p_{i,j}$ are obtained, the blocking and dropping probabilities for SUs can be calculated

Performance Metrics. Based on the aforementioned iterative algorithm, different performance metrics such as blocking probability, dropping probability and throughput, can be derived. An SU gets blocked if upon its arrival, all primary channels are occupied. In such case, the blocking probability, P_{b_1}, can be written as follows

$$P_{b_1} = \sum_{i=0}^{c_1} \lambda_2 p_{i,c_1-i}$$

If a PU arrives and transmits in the same channel that is already occupied by an SU, then an SU will be preempted. If there is no free primary channel to handoff, the SU will be dropped. In such case, the dropping probability, P_{d_1}, can be written as follows

$$P_{d_1} = \sum_{i=0}^{c_1-1} \lambda_1 p_{i,c_1-i}$$

The throughput T_1 can be defined as the average number of service completions for SUs per second. That is,

$$T_1 = \sum_{i=0}^{c_1-1} \sum_{j=1}^{c_1-i} j\mu_2 p_{i,j}$$

3.3 Opportunistic Spectrum Access with Backup Channels (OSAB)

This access scheme is different from the previous one in the way that an SU is accessing the available spectrum. An SU operates first in the primary channels and uses it as the operating channels. In case of the appearance of a PU, the SU should immediately handoff to the secondary channels. Therefore, the secondary channels are used as backup channels. In case that there are no secondary channels available, the SU handoff again to the primary channels. This is an extension for the work done in [11]. In [11], when a PU appears, an SU performs a handoff to the backup channels. If there is no backup channel, the SU is dropped. Figure 2 shows the transition diagram for this access scheme. The number of primary channels, c_1 is shared between PUs and SUs. The number of secondary channels, c_2 is used as backup channels in case of the sudden appearance of PUs. The process of spectrum access to the primary channels and secondary channels are modeled as a three-dimensional Markov chain. Therefore, states in the transition diagram are described by (i, j, k), where i is the number of primary channels used by PUs, j is the number of primary channels used by the SUs and k is the number of secondary channels used by SUs as backup channels. The state space S is given by

$$S = \{(i, j, k) \mid 0 \leq i \leq c_1, 0 \leq j \leq c_1 - i, 0 \leq k \leq c_2\}$$

Let $p_{i,j,k}$ be the steady-state probability distribution for a valid state $(i, j, k) \in S$. The state (i, j, k) can be moved to one of the following states depending on the arrival of PUs or SUs.

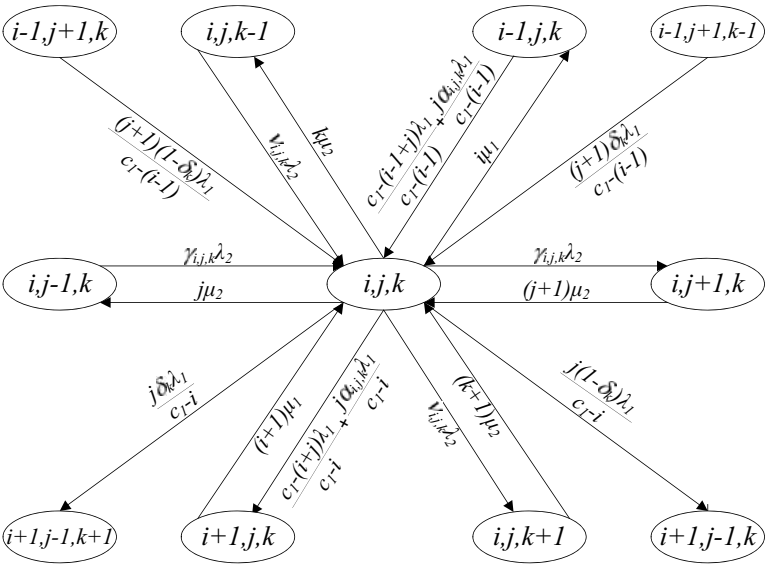


Fig. 2. Markov chain model for spectrum sharing with backup channels

1. State $(i+1, j, k)$. This case can be reached, if either one of the two following events happens:
   - a PU arrives and occupies a free channel which is not occupied by an SU with probability $\frac{c_1-(i+j)}{c_1-i}$. Therefore, the ongoing SU transmission will not be affected.
   - a PU arrives and occupies a channel which is utilized by an SU with probability $\frac{j\alpha_{i,j,k}}{c_1-i}$, whereas the indicator variable $\alpha_{i,j,k} = 1$, if $i+j < c_1$, $k = c_2$ and 0 otherwise. Therefore, the PU preempts the SU from this channel. Furthermore, the preempted SU performs a handoff to another free primary channel $i+j < c_1$, since there is no backup channel available $k = c_2$.
2. State $(i+1, j-1, k+1)$. This case happens, when a PU arrives and operates in the same channel that is occupied by an SU with probability $\frac{j\delta_k}{c_1-i}$, whereas the indicator variable $\delta_k = 1$, if $k < c_2$ and 0 if $k = c_2$. Therefore, the PU preempts the SU from this primary channel. Furthermore, the preempted SU performs a handoff to a backup channel since there is an available backup channel for the SU to complete his transmission (i.e. $k < c_2$).
3. State $(i+1, j-1, k)$. This case happens, when a PU arrives and operates in the same channel that is occupied by an SU with probability $\frac{j(1-\delta_k)}{c_1-i}$. Furthermore, there is no available backup channel for SU to complete its transmission (i.e. $k{=}c_2$). Therefore, the preempted SU will be dropped.
4. State $(i, j+1, k)$. This case happens, when an SU arrives and operates in a free primary channel (i.e. $i+j < c_1$), with probability $\gamma_{i,j,k}$, whereas the indicator variable $\gamma_{i,j,k} = 1$, if $i+j < c_1$, $k < c_2$ and 0 otherwise.

5. State $(i, j, k+1)$. This case happens, when an SU arrives and all primary channels are occupied (i.e. $i + j = c_1$), with probability $\nu_{i,j,k}$, whereas the indicator variable $\nu_{i,j,k} = 1$, if $i + j = c_1$, $0 < k < c_2$ and 0 otherwise. Therefore, the SU operates in a free secondary channels (i.e. $0 < k < c_2$).

Furthermore, the service completion for PUs, moves state (i, j, k) to states $(i-1, j, k)$. In addition, the service completion for SUs from one of the primary or secondary channels, moves state (i, j, k) to states $(i, j-1, k)$ or $(i, j, k-1)$, respectively. Based on the state transition diagram in Figure 2, the steady-state balance equation for $p_{i,j,k}$ is given as follows:

For $0 \leq i \leq c_1$, $0 \leq j \leq c_1 - i$ and $0 \leq k \leq c_2$.

$$\begin{aligned} B_{i,j,k}p_{i,j,k} = & \left(\frac{(c_1 - (i-1+j))}{c_1 - (i-1)} + \frac{j\alpha_{i,j,k}}{c_1 - (i-1)} \right) \lambda_1 p_{i-1,j,k} \\ & + \frac{(j+1)(1-\delta_k)}{c_1 - (i-1)} \lambda_1 p_{i-1,j+1,k} + \nu_{i,j,k}\lambda_2 p_{i,j,k-1} \\ & + \frac{(j+1)\delta_k}{c_1 - (i-1)} \lambda_1 p_{i-1,j+1,k-1} + \gamma_{i,j,k}\lambda_2 p_{i,j-1,k} \\ & + (k+1)\mu_2 p_{i,j,k+1} + (i+1)\mu_1 p_{i+1,j,k} \\ & + (j+1)\mu_2 p_{i,j+1,k} \end{aligned} \tag{2}$$

where the value of $p_{i,j,k} = 0$ for $i < 0$, $j < 0$ and $k < 0$ or $k > c_2$ or $i + j > c_1$. The value of $B_{i,j,k}$ is given as

$$B_{i,j,k} = \frac{(c_1 - i + j\alpha_{i,j})}{c_1 - i}\lambda_1 + (\gamma_{i,j,k} + \nu_{i,j,k})\lambda_2 + (j+k)\mu_2 + i\mu_1$$

Performance Metrics. Using the aforementioned iterative in section 3.2 with some modification, different performance metrics can be obtained such as blocking probability, dropping probability and throughput. An SU gets blocked if upon its arrival, all primary channels and secondary channels are occupied. In such case, the blocking probability, P_{b_2}, can be written as follows

$$P_{b_2} = \sum_{i=0}^{c_1} \lambda_2 p_{i,c_1-i,c_2}$$

If a PU arrives and transmits in the same channel that is already occupied by an SU, then an SU will be preempted. If there is no backup channel or free primary channel to handoff, the SU will be dropped. In such case, the dropping probability, P_{d_2}, can be written as follows

$$P_{d_2} = \sum_{i=0}^{c_1-1} \lambda_1 p_{i,c_1-i,c_2}$$

The throughput T_2 can be defined as the average number of service completions for SUs per second. Therefore,

$$T_2 = \sum_{i=0}^{c_1-1} \sum_{j=1}^{c_1-i} \sum_{k=0}^{c_2} j\mu_2 p_{i,j,k} + \sum_{i=0}^{c_1-1} \sum_{j=0}^{c_1-i} \sum_{k=1}^{c_2} k\mu_2 p_{i,j,k}.$$

4 Numerical Results

In this section, a comparison between OSA and OSAB is presented in term of blocking probability, dropping probability and throughput.

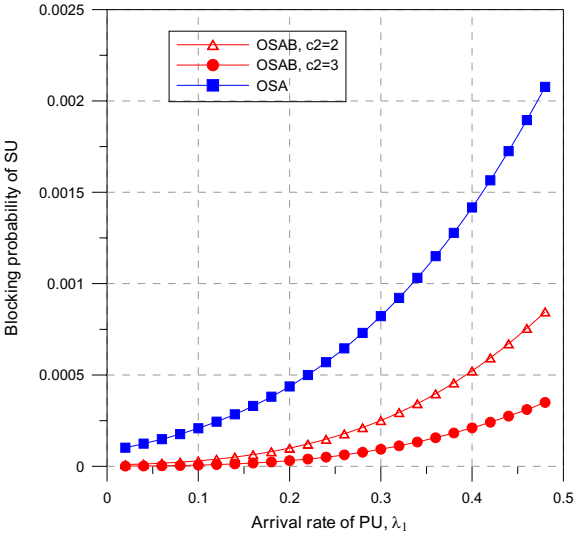


Fig. 3. Blocking probability for secondary user vs. arrival rate λ_1

Firstly, we show the impact of the number of secondary channels c_2 and the variation of the arrival rate λ_1 on the blocking and dropping probabilities of the SUs. The following operational parameters have been set as follows: $c_1 = 10$ channels, $\lambda_2 = 0.3$ SU/sec, $\mu_1 = 0.5$ PU/sec and $\mu_2 = 0.41$ SU/sec. As shown in Figure 3, the blocking probability for the SU in both schemes increases with respect to the arrival rate of PU; λ_1. This can be explained as follows: As λ_1 increases, the number of available channels that can be accessed opportunistically by the SUs reduces, which will lead to higher blocking probability for SUs. However, the blocking probability for OSAB is less than for OSA. This is intuitively clear, since OSAB increases the spectrum capacity for SUs by using the unlicensed band as backup. In Figure 4, the dropping probability for the classical OSA model is high compared to the one obtained from the OSAB model. The difference between the dropping probability for both models can be clarified as follows: in OSA model, as the arrival rate of PUs increases, the SU forced to perform a handoff to another free channel. This process continues till all the primary channels are occupied and hence the dropping probability for SU increases. In OSAB, when the arrival rate of PUs increases, the SU immediately performs a handoff to the backup channel from the secondary channels. The secondary channels are free from PUs and therefore, the SU will complete its transmission

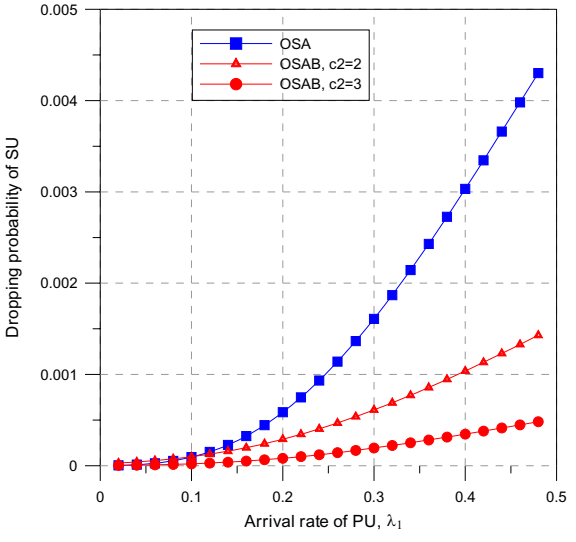


Fig. 4. Dropping probability for secondary user vs. arrival rate λ_1

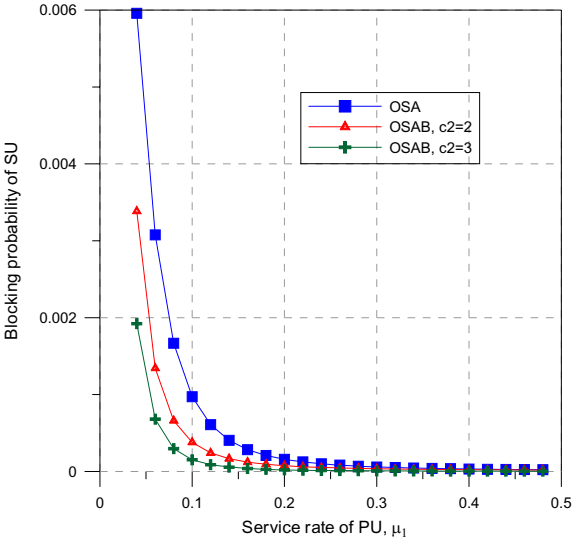


Fig. 5. Blocking probability for secondary user vs. service rate μ_1

and no need for spectrum handoff. If there are no free backup channels the SU will handoff again to the primary channels.

Secondly, we show the effect of the variation of the service rate μ_1 on the blocking and dropping probabilities of the SUs. It is obvious that when the service rate μ_1 increases, the channel holding time for the PU will be decreased

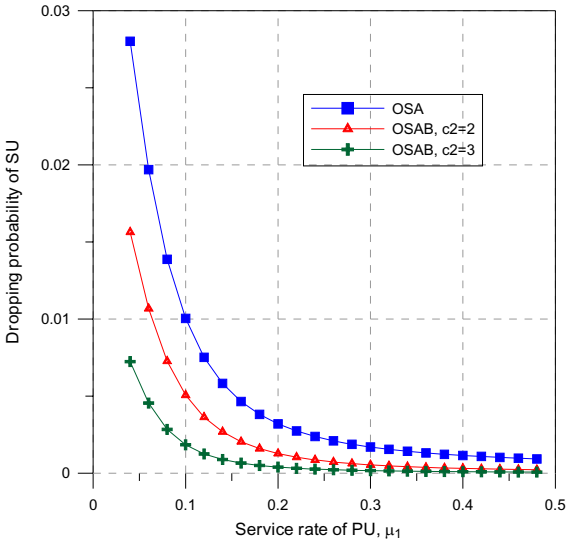


Fig. 6. Dropping probability for secondary user vs. service rate μ_1

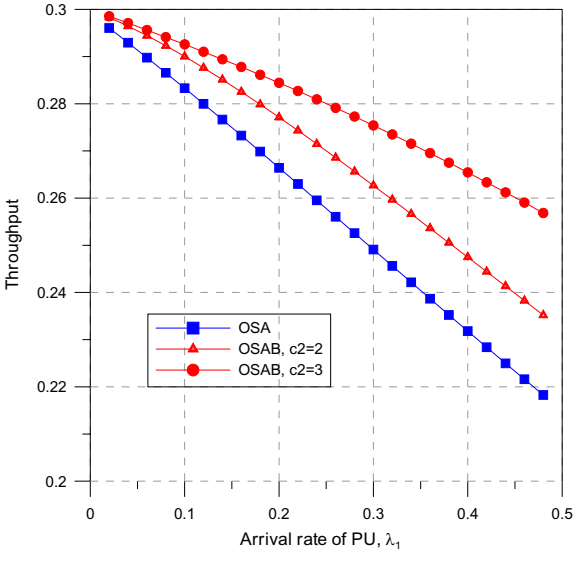


Fig. 7. Throughput for secondary vs. arrival rate λ_1

which leads to more primary channels being available. As a result, the blocking probability for the SU will be decreased since the arrival rate of PU is fixed λ_1=0.2 as shown in Figure 5. Figure 6 also illustrates the dropping probability for the SU. The figure shows that at a low service rate of PUs, the dropping

probability is high because the channel holding time for the PU is high. As the service rate of PUs increases, the dropping probability will decrease as the holding time for the PU is decreased. This is clear, because more SUs will be able to complete their transmission. Furthermore, the value of both blocking and dropping probabilities for OSAB, compared to OSA, is reduced as a reason of increasing the spectrum capacity for the SU.

Finally, Figure 7 depicts the throughput for SUs with variation of the arrival rate λ_1 and $\lambda_2 = 0.5$ SU/sec. At low traffic load for the PU, the throughput is high and when the arrival rate increases, the throughput decreases. This can be explained as follows: at low arrival rate for the PU, more available channels will be available for SUs and therefore the dropping probability for SUs will be decreased which will lead to an increase at the throughput. When the arrival rate of PUs increases, the dropping probability will be increased and as a result the throughput will decreased.

5 Conclusion

In this paper, the performance of SUs in a heterogeneous environment of licensed and unlicensed channels has been evaluated analytically. The average spectrum usage for the SU is increased by employing licensed channels as operating channels and unlicensed channels as backup channels. The results show a significant improvement of our model compared to the classical OSA in term of blocking and dropping probabilities for SUs. Also, the throughput is increased due to the proposed scheme. In future work, a validation for those metrics will be done through a simulation.

References

1. FCC Spectrum Policy Task Force: Report of the spectrum efficiency working group. Federal Communications Commission, Technical Report 02-155 (November 2002)
2. Mitola, J., Maguireet, G.Q.: Cognitive Radios: Making Software Radios more Personal. IEEE Personal Communications 6(4), 13–18 (1999)
3. Mitola, J.: Cognitive radio: An integrated agent architecture for software defined radio. PhD Dissertation, Royal Inst. Technol. (KTH), Stockholm, Sweden (2000)
4. Akyildiz, I.F., Lee, W.Y., Vuran, M.C., Mohanty, S.: NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey. Computer Networks Journal 50, 2127–2159 (2006)
5. Akyildiz, I.F., Lee, W.Y., Vuran, M.C., Mohanty, S.: A Survey on Spectrum Management in Cognitive Radio Networks. IEEE Communications Magazine 46(4), 40–48 (2008)
6. The Next Generation Program, http://www.darpa.mil/sto/smallunitops/xg.html
7. Zhu, X., Shen, L., Yum, T.P.: Analysis of Cognitive Radio Spectrum Access with Optimal Channel Reservation. IEEE Communications Letters 11(4), 304–306 (2007)

8. Xing, Y., Chandramouli, R., Mangold, S., Shankar, N.S.: Dynamic Spectrum Access in Open Spectrum Wireless Networks. IEEE Journal on Selected Areas in Communications 24(3), 626–637 (2006)
9. Tang, P.K., Chew, Y.H., Ong, L.C., Haldar, M.K.: Performance of Secondary Radios in Spectrum Sharing with Prioritized Primary Access. In: Proceedings of Military Communications Conference (MILCOM 2006), pp. 1–7 (2006)
10. Wang, L., Chen, A., Wei, D.S.: A Cognitive MAC Protocol for QoS Provisioning in Overlaying Ad Hoc Networks. In: Proceeding of 4th IEEE Consumer Communications and Networking Conference 2007 (CCNC 2007), pp. 1139–1143 (2007)
11. Kalil, M.A., Liers, F., Volkert, T., Mitschele-Thiel, A.: A Novel Opportunistic Spectrum Sharing Scheme for Cognitive Ad Hoc Networks. In: 5th Workshop on Mobile Ad-Hoc Networks (WMAN 2009), Kassel, Germany, March 2009. Published online on Electronic Communications of the EASST, vol. 17 (2009)

How Would Ants Implement an Intelligent Route Control System?

Hamid Hajabdolali Bazzaz[1,2] and Ahmad Khonsari[1,2]

[1] ECE Department, University of Tehran, North Kargar Ave., Tehran, Iran
[2] School of Computer Science, IPM, Niavaran Sq., Niavaran, Tehran, Iran
h.hajabdolali@ece.ut.ac.ir, ak@ipm.ir

Abstract. Multihoming, the connection of a stub network through multiple Internet Service Providers (ISPs) to the Internet, has broadly been employed by enterprise networks as a sort of redundancy technique to augment the availability and reliability of their Internet access. Recently, with the emergence of Intelligent Route Control (IRC) products, IRC-capable multihomed networks dynamically select which ISPs' link to use for different destinations in their traffic in a smart way to bypass congested or long paths as well as Internet outages. This dynamic traffic switch between upstream ISPs is mostly driven by regular measurement of performance metrics such as delay, loss ratio, and available bandwidth of existing upstream paths. However, since IRC systems are commercial products, details of their technical implementation are not available yet. Having the incentive to delve into these systems deeply, in this paper, we employ traditional ant colony optimization (ACO) paradigm to study IRC systems in that domain. Specifically, we are interested in two major questions. Firstly, how much effectively does an ant based IRC system switch between upstream links in comparison to a commercial IRC system? Secondly, what are the realistic underlying performance metrics by which ants pick the path to a food source (destination network) in a multihomed colony? Through extensive simulations under different traffic load and link reliability scenarios, we observe that ants perform well in switching between available egress links. Moreover, delay of paths is not the only criterion by which ants select the path; instead, through their intuitive ACO paradigm, they tend to choose the path with a better performance in terms of both delay and loss ratio.

Keywords: Intelligent route control, ant colony optimization.

1 Introduction

Multihoming has been widely used as a redundancy technique to provide stub enterprise networks with a higher level of availability and reliability in their Internet access. In this practice, each stub network is connected to the Internet through several Internet Service Provider (ISP) upstream links rather than a single one. In the most straightforward deployment configuration, one of the ISPs is picked as the primary Internet provider while the others are just used as backup providers and only in the case of failures in the primary connection (Fig. 1).

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 29–41, 2009.

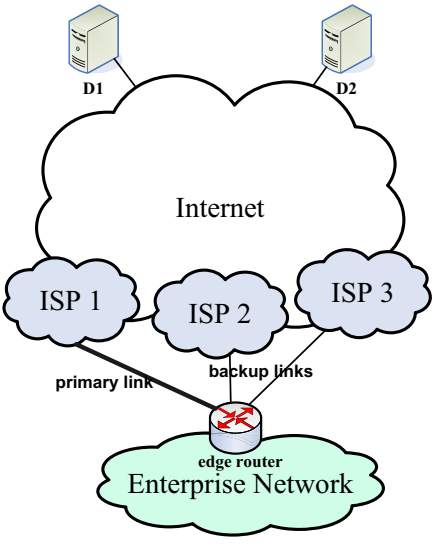


Fig. 1. A multihomed network with three upstream ISPs

Recently by the emergence of Intelligent Route Control (IRC) products [1]-[3], instead of using only one of the upstream links at each time, all egress links are utilized and the IRC edge router dynamically selects the best upstream ISP for every major destination of the network traffic. This is typically done driven by some performance criterion such as delay and loss ratio. As a result, the proliferation of IRC systems by stub networks is not only for bypassing Internet outage but also to have their traffic experience a low latency and packet loss.

There are just few recent studies on the potential performance/cost benefits of IRC systems [4]-[7]. However, since IRC systems are commercial products, only the basic implementation guidelines are discussed thus far. In this paper, we apply traditional ant colony optimization (ACO) paradigm to study these systems deeply. Because, as we observe throughout the paper, the way ants select their path to a food source using their heuristic pheromone-basis ACO paradigm is very similar to commercial IRC systems deployment. We translate the IRC problem into ant colony domain where ants are seeking the best egress path between their multihomed colony and source food destinations. Specifically, we are interested in two primary questions. Firstly, how much effectively does an ant based IRC deployment pick upstream links in comparison to a commercial IRC system? Secondly, what are the realistic underlying performance criterions by which ants select the path to a destination (food source) in a multihomed colony? We address these questions throughout this paper.

The rest of the paper is structured as follows. Section 2 reviews the commercial IRC implementation guidelines from the literature. In Section 3, we translate the IRC problem into ant colony domain and argue about different design decisions we need to make. Performance, convergence time and criterion metric of the proposed ant based IRC system are evaluated in Section 4 and finally we conclude the paper in Section 5.

2 Commercial IRC Systems

There are few works on performance evaluation of IRC systems such as [4]-[11]. More recently, in [5][6] Akella *et al.* emulate an IRC system on Akamai content distribution network and argue that multihoming can bring about a considerable amount of benefits in terms of both availability and performance. As for the degree of multihoming, their results state that IRC has the potential to achieve an average performance benefits improvement of 25% or more for a 2-multihomed stub network. The authors also show that typically having up to four upstream providers is enough to gain the full benefit out of multihoming.

In previous works [12], an IRC system behavior is typically modeled as a periodic five-stage process, namely, idle, measurement, performance estimation, routing decision, and path switching. The whole period is T_R seconds. There is an optional idle stage in the beginning of every cycle in which the system doesn't probe the destinations to reduce the overhead of the routing probes. Then, in the measurement stage which is T_M seconds, different metrics of the existing upstream paths (e.g., delay, loss ratio and available bandwidth) are collected through sending N_p number of *probing* packets. After receiving the acknowledgements of these probes, the IRC system estimates a level of performance for each candidate path by a hybrid metric typically consisting of delay and loss ratio of the paths. In the routing decision stage, one of the paths is selected as the best upstream path based on the previous stage performance estimations. And finally, the IRC system routes the traffic through the chosen egress link at least during the entire coming routing period. Notice that performance estimation, routing decision, and path switching stages are almost immediate tasks because they only include simple calculation and forwarding table (memory) update. Our proposed deployment of the IRC system follows this basic 5-stage process. We discuss the details of the ant based approach in the following Section.

3 Ant Based IRC System

Ant Colony Optimization (ACO) initially was proposed by Marco Dorigo in 1992 in his PhD thesis [13]. The aim of the first algorithm was searching for an optimal path in a graph based on the behavior of ants seeking a path between their colony and a food source [14]-[16]. Intuitively, ants do so by wandering around randomly on different paths while laying down *pheromone* trail, a chemical substance they use to form an indirect communication with each other. As soon as an ant finds a path to a source food, it comes back to the colony while still putting down pheromone on its way back to the home colony. Ants greedily believe that paths which are traveled more (by their companions) and, thus have more pheromone are more likely to be leading to source food. Thus, while an ant meanders randomly to find a short path to a food source, the chance it follows a specific existing path is somehow related to the amount of pheromone its senses on that specific path. Over time, however, the pheromone trail evaporates [17]. The more time it takes for an ant to travel down a specific path and back to the home again, the more time the pheromones have to evaporate. Yet, a short path gets marched over faster, and thus its pheromone density remains high even though its pheromones evaporate. As a result, the problem of

finding the best path (e.g., shortest one) which the ants are trying to solve naturally is indeed heuristically solved by this simple approach they follow. Thus far imitating ants' behavior, diverse numerical problems are solved by employing ACO. See [15] for a complete survey.

In this section, after this brief overview on ACO, we formulate the IRC problem as an ACO problem. Specifically, we are to denote the notations of ACO we employ in a typical IRC system, namely:

(3.1) What are the correspondent elements of IRC in our ant based model?
(3.2) How much pheromone do ants lie down on the path they travel? Is this amount constant or it may change dynamically?
(3.3) How long does it take for pheromones to evaporate? In other words, what is the pheromone evaporation function?
(3.4) How exactly do ants select the path they march on?

In the coming subsections we address these questions.

3.1 Ant Colony Model

We consider a simple multihomed network with an IRC capable edge router which major traffic is destined to several probably big networks which are given a priori. In our model, the source IRC-equipped stub network represents the ant colony while each of its major destinations is like a food source to which ants are trying to find the best path (Fig. 2). In the ant based IRC system, ants play the role of the routing probes the edge router periodically transmits through its upstream links in order to obtain performance metrics of each of its paths to a specific destination. Both systems (ant based IRC and commercial one) have the same objective; calculating the best path to

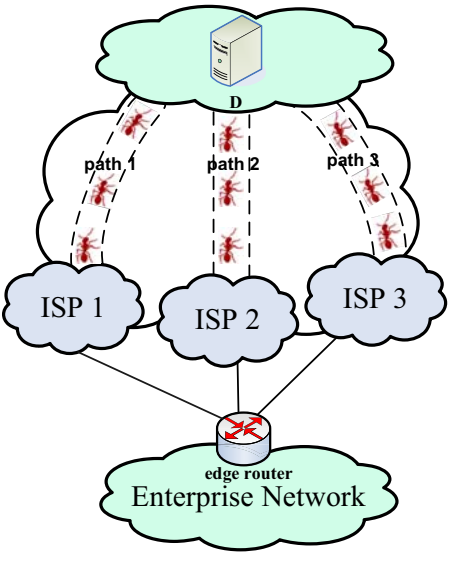


Fig. 2. An ant colony IRC system

the major destinations of the stub network. There are actually two choices for the rate by which new ants march from the source to each destination. First, we may consider this rate as the traffic rate between the stub network and that destination. The alternative is to map this rate only to that of probing packets. Indeed, there are no significant differences between these two schemes, but as the principle purpose of the IRC system is finding out the best path per destination through probing, we take the second option. As a result, in our model the rate of data packets between the source and destination is not of importance given the fact that solely probing ants are trying to calculate the best path. This roughly means that we have two sorts of ants: the elite type (routing probes packets) which generates pheromone while wandering and the normal ants (data packets) which solely traverse the best path picked by their elite companions. Unless otherwise specified, we are referring to the elite ants simply by ants throughout the paper.

3.2 Pheromone Creation Formulation

As the ants march on different upstream links, they lay down pheromone trails on the path they travel. Eventually, this would result in the augmentation of the pheromone value associated with good paths (i.e., shorter paths) to encourage more ants to pick them, and decrease in that of bad paths (i.e., longer paths) through pheromone evaporation. Here, we formulate the pheromone update scheme we use throughout this paper by adapting the *aging* phenomena in the ACO [14]. Aging phenomena simply states that an older ant may produce less pheromone as a result of its decrepitude. It is one of the methods used to control the amount of pheromone of the whole system in the ACO. Thus, by adopting this fact, we assume that (i) the initial amount of pheromone an ant lies down on the paths it travels on is a constant value K_0; (ii) ant's pheromone generation rate decreases exponentially while it becomes older as the result of the aging phenomena. Specifically, the amount of pheromone an ant puts on a path at the age of t (seconds) is $K_0 e^{-t}$. Let $\tau_{s,i}(.)$ represent the amount of pheromone at egress point of path i of the enterprise source node s. Thus, if at time t_0 an ant a with the age of $age(a)$ marches on the egress point of path i of the IRC equipped enterprise network s, we have:

$$\tau_{s,i}(t_0 + \varepsilon) = \tau_{s,i}(t_0) + K_0 e^{-age(a)} \tag{1}$$

in which $0 < \varepsilon << 1$ is a very small number.

3.3 Evaporation Rate Formulation

As explained earlier, to reduce the impact of past experience, an approach called evaporation [17] is typically applied in ACO. Evaporation avoids pheromone concentration in optimal paths from being excessively high and preventing ants from exploring other (new or better) alternatives [14]. Furthermore, with employing evaporation it is possible to switch between paths in case of dynamic changes in the optimal solution of the problem. Indeed, this is exactly what happens in the ant based IRC and we discuss this more later. Here, we formulate the pheromone evaporation paradigm we use in our ant based IRC. We assume that the initial K_0 units of pheromones an ant put

down on the path are evaporated T_E seconds later. This roughly results in a pheromone evaporation rate of $\frac{K_0}{T_E}$.

Now, by considering aforementioned pheromone generation and evaporation formulas, we can analytically calculate the pheromone update function $\tau_{s,i}(.)$, recursively by adding up the pheromone value changes during interval $T=[t,t+\Delta t]$ on each path. Specifically, we have:

$$\tau_{s,i}(t+\Delta t)=[\tau_{s,i}(t)(1-\Delta t\frac{K_0}{T_E})]^+ + \sum_{a\in i \text{ in } T}[K_0 e^{-age(a)}(1-(\Delta age(a))\frac{K_0}{T_E})]^+ \quad (2)$$

where $[z]^+=\max\{0,z\}$. In this formula, the first term corresponds to the residual amount of the initial value of pheromone on the path at time *t* after decreasing evaporated amount out of it. The second term is the residual amount of pheromone added to the path as the result of ants walking on it during the interval *T*. Note that the summarization is over all the ants marching on the egress point of path *i* in the time period *T*. $age(a)$ represents the age of ant a exactly at the moment it is on the path and $\Delta age(a)$ denotes its age increase from the initial time it is on the egress point till $t+\Delta t$.

3.4 Ants Routing Strategy

In this subsection we bring up the routing strategy ants utilize to reach source foods and then travel back to their colony. We distinguish between three cases as follow:

1) Egress upstream link: Ants select the egress upstream link attached to their source colony in a probabilistic nature based on the amount of pheromone they sense at each egress point at time they want to wander out of their colony. Specifically, if we denote the probability that an ant selects the egress upstream link *i* attached to the source colony *s* at time *t* by $p_{s,i}(t)$, we have:

$$p_{s,i}(t)=\frac{\tau_{s,i}(t)}{\sum_i \tau_{s,i}(t)}. \quad (3)$$

2) Path from a specific egress link to a destination food source: Once an ant has chosen the egress upstream link, it has to find out the path to the source food through that link. There are two choices for this path selection strategy. Firstly, to do so similar to egress link selection in a probabilistic nature through some ant based optimization schemes (e.g., following the pheromone trails). Secondly, to utilize the current available forwarding tables of each intermediate node to reach the destination. We stress that in an ant based IRC system, the objective is solely picking the egress link to a specific destination rather than the whole path which is already plain with the aid of underlying routing algorithms. Thus we select the second alternative. One may note that the first choice would work as well with some additional overhead in the convergence time of the ACO. In other words, here, we

assume that ants have already solved the traditional problem of finding the best path from each node in the given network to the source foods and our focus is on IRC nature of the problem.

3) Path from the destination to the source colony: Once an ant reaches a source food, it can simply travel back on the path it has already found between the source and destination to get back to its home colony. As a matter of fact, this travel back methodology has been traditionally used in ACO [13].

4 Experimental Results

This section presents a performance evaluation of our proposed ant colony based IRC system. Specifically, we are interested in two aforementioned questions. Firstly, how much effectively does the ant based system switch between paths in comparison to a commercial IRC system? Secondly, what are the underlying performance criterions by which ants switch between paths?

4.1 Simulation Setup

Our simulations are performed with a customized simulator, implemented in C++, which precisely model the network topology and traffic at packet level. Unless otherwise specified, we set simulation parameters like this: $K_0 = 1$, $T_E = 3$, $T_R = 2$, $T_M = 0.8$, and N_p, the number of probing packets in each measurement period, is selected 30. We have conducted extensive simulations to study effects of each of these key parameters on the results and then selected aforementioned values for each parameter. In subsection 4.2, we review how we derived these values for the simulation parameters.

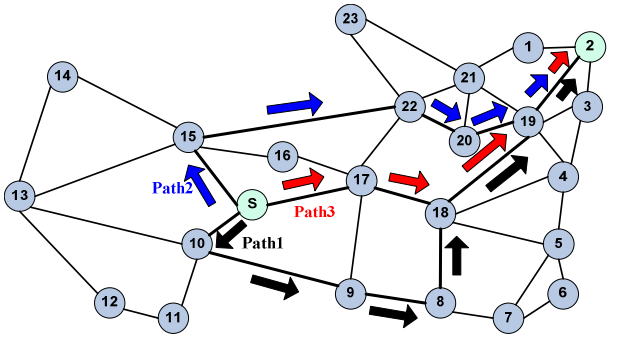


Fig. 3. The simulated network

The test network used in our experiments consists of a 23-node 38-link network [19] depicts in Fig. 3 All links are 1 MB/s with the same routing metric cost. The background traffic consists of a number of random flows between nodes in the topology. The node pairs, start time, duration, and rate of these flows are selected randomly according to a uniform distribution. In the simulated network, source node S is multihomed by three different upstream links, namely to nodes 10, 15, and 17. We

pick node 2 as the major destination (food source) of this node due to the appropriate path diversity of selected upstream nodes to this destination. In particular, given the same cost for every link in the topology, any shortest-path routing algorithm (such as OSPF) picks the following paths as the shortest path between aforementioned upstream nodes to node 2:

Path1: <S, 10, 9, 8, 18, 19, 2>
Path2: <S, 15, 22, 20, 19, 2>
Path3: <S, 17, 18, 19, 2>

Here, we have chosen the node with lower id in case of any ties. For instance in the Path1, between two available paths with the same cost from intermediate node 22 to node 2 (<22, 20, 19, 2> and <22, 21, 1, 2>), the path through node 20 is selected rather than that of node 21.

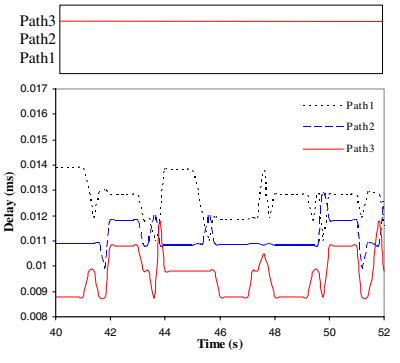


Fig. 4. Comparison of delay of paths at different time (bottom graph) along with chosen path (top graph) in the commercial IRC in scenario 1

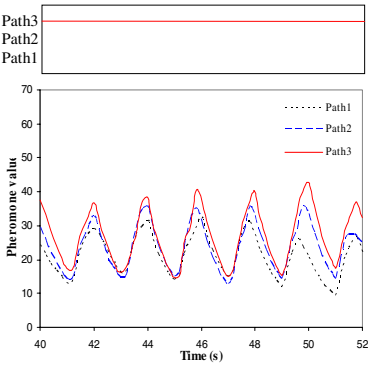


Fig. 5. Comparison of pheromone of paths at different time (bottom graph) along with chosen path (top graph) in the ant based IRC in scenario 1

We explore the behavior of the ant based system and compare its effectiveness with a commercial delay based IRC system, i.e., an IRC system which path switching decisions are made solely on the basis of delay metric of the paths, in three different scenarios. In the first scenario, the underlying network traffic load on the paths is low and the paths are not congested as well as reliable and stationary. In other words, there are no temporal congestion, or delay oscillation and bursty packet losses. In the second scenario, we put a temporal congestion event on the Path1 which brings about a longer delay on this path. We then study the reaction of the systems to this delay increase. In the final scenario, we examine the impacts of unreliable links, e.g., lossy ones in a wireless like environment, on the efficiency of the systems.

4.2 Performance Evaluation under Stationary Conditions

Assuming the same propagation delay for all the links in the simulated network, Path3 has the lowest minimum round trip time (RTT) due to having minimum number of intermediate nodes between source node S and destination node 2; after that comes Path2 and finally Path1 which has the longest native delay. In this section, we examine the stationary scenario in which the underlying network paths are under low load and they are stable in terms of delay, packet loss and congestion.

Fig. 4 depicts the delay of the paths during a part of simulation time interval. The top graph in this figure shows corresponding chosen path of the delay based IRC system in each interval. As the switching decisions are made exactly every T_R seconds (2 sec in this case) based on the average measured delay of the latest measurement interval T_M (0.8 sec in this case), at each point the nearest measured delay is shown in the graph. This explains, the reason the curves are straight for an interval of T_R - T_M (1.2 seconds in this case) every T_R seconds. According to the stable traffic load of this scenario, Path3 always has the lowest delay on average; Path2 comes next while Path1 has the longest delay. Thus, the delay based IRC system has always selected egress link of Path3 (the top graph) in this scenario.

On the other hand, Fig. 5 shows the ant based IRC system path switching events along with correspondent switching metric, i.e., amount of pheromone on each egress link. As it is seen in this figure, the system works effectively and similar to the commercial delay based system in terms of selecting the best upstream link during every iteration.

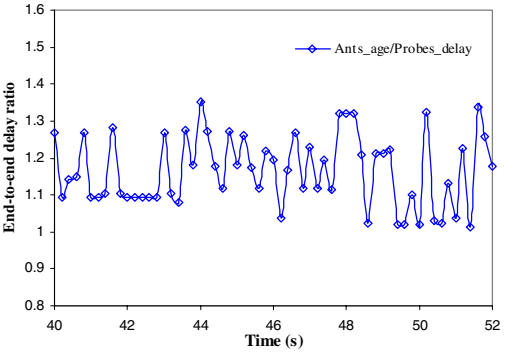


Fig. 6. Comparison of ants and routing probes end-to-end delay at different time for Path1 in scenario 1

Another observation to make is regarding the ant age in the ant based IRC system and its correspondent in the commercial IRC system, i.e., packets end-to-end delay. Fig. 6 shows the ratio of this end-to-end delay measurement of the ant based system to that of the commercial IRC system for Path1. As it is seen in this graph, ants always have greater end-to-end delay than IRC probing packets. The reason behind this is that although both ants and routing probes follow the same path to the destination, yet on traveling back to the source network, routing probes come back on the shortest path

between the destination and source network (Path3 in this case) while ants travel back on exactly the same path (Path1 in this case) they have found from their source colony to the destination source food. This clearly led to a higher end-to-end delay for the ants due to the larger delay ants experience on their path from destination to their home.

A discussion on the selection of the system key parameters is due here. First, note that K_0, the initial amount of pheromone ants lay down while marching around, doesn't have a critical impact on the path selection. Indeed, increasing/decreasing this parameter would increase/decrease the total amount of pheromone per iteration yet switching decisions are not altered. However, evaporation duration time, T_E, has a significant effect on the convergence duration of the ant based IRC system. As an example, in Fig. 5, by choosing $T_E = 3$, we observe that at the end of each measurement period, the ant are able (e.g., have enough time) to find the best path correctly. However, if we reduce T_E, to 2 seconds or so, the convergence duration may be longer and ants wouldn't find the best path until the second iteration due to the rapid rate of pheromone evaporation which results in a lower difference between the amount of pheromone on each path. The lower difference of pheromone values in turn makes finding the best path a problematic issue for the ants as a result of probabilistic nature of ants' movement. On the other hand, a bigger value of T_E would make the initial convergence duration lower while bringing about a longer convergence time for the consequence path switching events in the case of sudden congestion on the selected path. Furthermore, it is evident that $T_E \geq T_R$ is a necessary constraint. With these in mind, $T_E = 3$ is the best selection as expected and seen in our results. We would suggest $T_E \approx 1.5 \times T_R$ in a more generalized scheme.

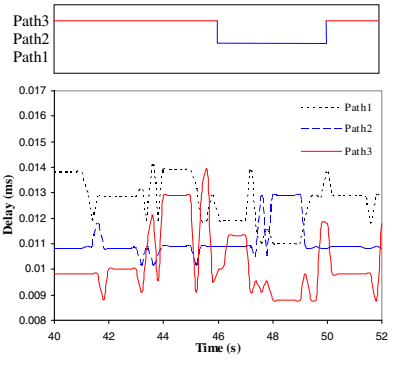


Fig. 7. Comparison of delay of paths at different time (bottom graph) along with chosen path (top graph) in the commercial IRC in scenario 2

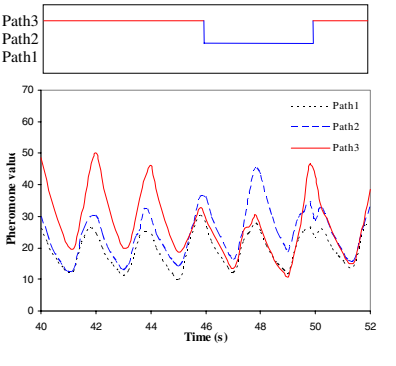


Fig. 8. Comparison of pheromone of paths at different time (bottom graph) along with chosen path (top graph) in the ant based IRC in scenario 1

As for T_R and T_M parameters, the bigger the T_M is, the faster ants tend to find the best path. Thus, this actually imposes a trade-off between the amount of overhead we put into the network for probing the destinations through each available path and the speed of convergence which satisfy us. Furthermore, for faster response in case of network congestion, the idle period T_R -T_M within each iteration may be minimized or

even avoided. And, finally, it is important to note that the number of routing probe packets, N_p, is of importance in the effectiveness of the system. Choosing a low value for N_p, affects the correctness of the measurements and convergence time of the ant based system while a large value for N_p imposes too much useless overhead. In addition, an approximate upper bound on N_p for each path can be driven to assure that the probes are acknowledged by the destinations *on time*, i.e., before the next switching decision event. If we represent N_p on egress link l by $N_{p,l}$, we have:

$$\forall_l \frac{T_M}{RTT_l} \succ N_{p,l} \tag{4}$$

where, RTT_l denotes the average round trip time on the path correspondent to l. Considering the path with the maximum RTT, we can calculate the upper bound on N_p. For instance, in our experiments, $T_M = 0.8$ and $\max\{RTT\} \approx 0.2\,ms$ results in an upper bound of 40 for N_p. However, we notice that $N_p = 30$ is big enough in terms of convergence speed.

4.3 Performance Evaluation under Congested Links

In this scenario, to study the effectiveness of the ant based IRC system in relation to that of a delay based one in case of path switching events, we introduce a congestion event at Path3 by CBR cross traffic between nodes 17 and 5, i.e., <17, 18, 5>, during the time interval [44, 48].

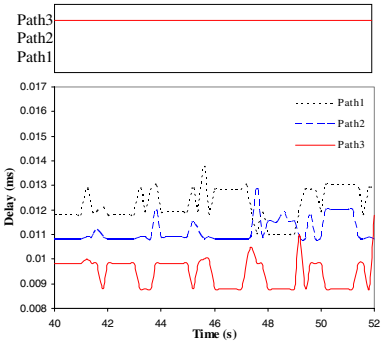


Fig. 9. Comparison of delay of paths at different time (bottom graph) along with chosen path (top graph) in the commercial IRC in scenario 3

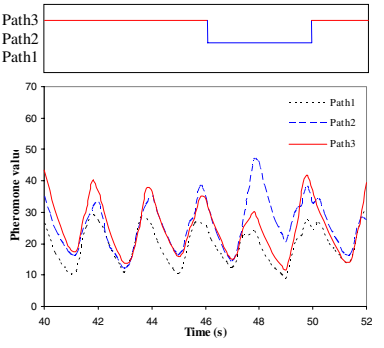


Fig. 10. Comparison of pheromone of paths at different time (bottom graph) along with chosen path (top graph) in the ant based IRC in scenario 3

Fig. 7 and Fig. 8 depict the path switching events of the IRC systems along with their switching metric. We observe that both systems recognize the congestion event on Path3 during interval [44, 46] and on the next switching event in 46 both switch to the best alterative after that path, Path2. Besides, once Path3 has recovered from temporal congestion in 48, both systems have switched back to that path on the next switching event at time 50.

4.4 Performance Evaluation under Lossy Links

In order to analyze effects of lossy links in a wireless like environment, we assume that link <17, 18> losses packets with probability 40% during time interval [44, 48] (See Fig. 11). Fig. 9 and Fig. 10 show the path switching events of the ant based and delay based IRC systems along with their switching metric. We make a few observations. First, the ant based system is still working effectively. Surprisingly, in time period [44, 46] ants have recognized the poor reliability of the link(s) on Path3 and on the next switching event in time 48, they have switched to the Path2 as it is the best alternative considering low delay after Path3. However, the delay based system hasn't figure out this lossy behavior as it only considers the delay metric of the probes which arrive to it for making its decisions. Furthermore, again ants have switched back to Path3, once the lossy behavior of this path is finished.

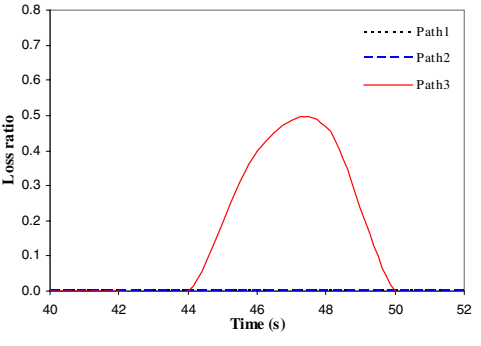


Fig. 11. Comparison of loss ratio of paths at different time in scenario 3

In sum, the results of experimental evaluation of this section evidently suggest that (i) ants efficiently choose the best egress links in comparison to a commercial IRC system. Indeed, all of the switching events of a delay based IRC system are detected and followed by the ants as well. (ii) The path ants heuristically select has low latency and also low packet drops. In other words, ants choose paths with a hybrid metric of delay and loss ratio.

5 Conclusion

The emergence of Intelligent Route Control (IRC) products has enabled the augmentation of the reliability, availability and also the end-to-end performance of multi-homed stub networks' connection to the Internet. Leveraging ideas from ant colony optimization (ACO), this paper presents an ant based deployment of IRC systems to see how ants would intuitively implement such systems. Using extensive simulation experiences under different network traffic load and link reliability scenarios, we obtain two important results. First, ants perform quite efficiently in selecting the best

upstream path in comparison to commercial IRC systems. Second, we observe that the path ants tend to pick at each time through their heuristic pheromone-basis ACO paradigm is actually the most optimized upstream path in terms of a hybrid metric of both delay and loss ratio.

References

1. InterNAP, Premise-Based Route Optimization
2. Route Science, Adaptive Networking Software
3. Cisco Systems, Optimized Edge Routing (OER)
4. Akella, A., Seshan, S., Shaikh, A.: Multihoming Performance Benefits: An Experimental Evaluation of Practical Enterprise Strategies. In: Proc. USENIX, Boston, MA (June 2004)
5. Akella, A., et al.: On the Performance Benefits of Multihoming Route Control. IEEE/ACM Transactions on Networking (2008)
6. Akella, A., et al.: A Measurement-Based Analysis of Multihoming. In: Proc. ACM SIGCOMM, Karlsruhe, Germany (August 2003)
7. Goldenberg, D., Qiu, L., Xie, H., Yang, Y., Zhang, Y.: Optimizing Cost and Performance for Multihoming. In: Proceedings of ACM SIGCOMM (2004)
8. Dai, R., Stahl, D.O., Whinston, A.B.: The economics of smart routing and QoS. In: Proceedings of the Fifth International Workshop on Networked Group Communications (2003)
9. Schreck, G., Rustein, C., Porth, M.: The end of the private WAN. Forrester Brief (March 2002)
10. Sevcik, P., Bartlett, J.: Improving user experience with route control. Technical Report NetForecast Report 5062. NetForecast (2002)
11. Guo, F., Chen, J., Li, W., Chiueh, W.: Experiences in Building A Multihoming Load Balancing System. In: Proceedings of IEEE INFOCOM (2004)
12. Gao, R., Dovrolis, C., Zegura, E.W.: Avoiding oscillations due to intelligent route control systems. In: Proc. IEEE INFOCOM (April 2006)
13. Dorigo, M.: Optimization, Learning and Natural Algorithms. PhD thesis. Politecnico di Milano, Italy (1992)
14. Sim, K.M., Sun, K.M.: Ant colony optimization for routing and load-balancing: survey and new directions. IEEE Transactions on Systems, Man and Cybernetics 33(5), 560–572 (2003)
15. Bonabeau, E., Dorigo, M., Theraulaz, G.: Inspiration for optimization from social insect behavior. Nature 406, 39–42 (2000)
16. Dorigo, M., Caro, G.D.: The ant colony optimization metaheuristic. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 11–32. McGraw-Hill, New York (1999)
17. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. Theoretical Computer Science 344(2-3), 243–278 (2005)
18. Schoonderwoerd, R., Holl, O., Bruten, J., Rothkrantz, L.: Ant-based Load Balancing in Telecommunications Networks. Tech. Rep., Hewlett Packard Lab. Bristol, U.K., HPL-96-35 (1996)
19. Zarifzadeh, S., Yazdani, N.: Joint Resource Conserving and Load Distributing Approach for Routing of Survivable Connections. Elsevier Computer Communications 31(14), 3384–3393 (2008)

User Access to Popular Data on the Internet and Approaches for IP Traffic Flow Optimization

Gerhard Hasslinger[1], Franz Hartleb[1], and Thomas Beckhaus[2]

[1] T-Systems Enterprise Services GmbH
Deutsche-Telekom-Allee 7
D-64295 Darmstadt, Germany
{gerhard.hasslinger,franz.hartleb}@telekom.de
[2] Deutsche Telekom Netzproduktion GmbH
Heinrich-Hertz-Str. 3-7
D-64295 Darmstadt, Germany
thomas.beckhaus@telekom.de

Abstract. Content delivery networks (CDN) and peer-to-peer systems currently account for the transport of a major portion of the Internet traffic. We compare the efficiency of both approaches, which are based on overlay structures within the network (CDN) or on the terminals of the users (P2P). Random source selection schemes in peer-to-peer protocols often chose to download content from somewhere across the globe although it may be available in the proximity, which leads to unnecessary high traffic load on inter-domain links. For content delivery networks, the distances from source to destination depend on the number and locations of servers involved in the CDN. Recent proposals to improve local exchange of popular data in the Internet are discussed with different implications for network resource efficiency, service provisioning and usage.

Keywords: Content delivery, access pattern, Zipf law, application layer traffic engineering.

1 Introduction

Overlays of various types are used to bridge heterogeneous networks for seamless end-to-end services. Current networking trends towards fixed/mobile convergence make them even more relevant for integrating different technologies with Internet access. A major advantage of overlays is their ability to provide new services or extend existing ones on top of and independent of an underlying infrastructure.

Peer-to-peer (P2P) networks establish overlays on the terminal equipment of the users, offering global services at a minimum of own network infrastructure for the provider. Fast delivery of large volumes of content within globally scattered communities is a main strength of the peer-to-peer principle, with scalability and adaptability for heterogeneous access [5][29]. Peer-to-peer networks are highly efficient for several purposes due to their ability

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 42–55, 2009.

- to exploit vacant resources (data, storage, computation power, bandwidth) distributed over user equipment,
- to adapt to varying demands with scalability for huge communities, including dynamic flash crowds,
- to embed support for search and replication of data for predefined demands in a self-organizing way,
- to build and manage overlays without or at low cost for network and server infrastructure.

On the other hand, transmission paths for P2P data exchange often span around the globe although popular data is most often available in the near of a requesting peer. In contrast to P2P networks, content delivery networks (CDN) form overlays based on distributed server farms, which are known to do a good job in optimizing transmission paths and corresponding delay by delivering data from a server close to each requesting location [8][22][30].

Network providers have to deal with overlay traffic in the planning and upgrading process, while they employ mechanisms for load balancing on their platforms [13][20]. In this way, inefficient routing on higher layers can be partly reduced by optimization on lower layers. Traffic engineering in overlays and on the application layer may offer additional optimization capabilities, but on the other hand may lead to increased overhead and cross-layer inefficiency by repeating similar and overlapping functions on higher layers or, in the worst case, by employing functions jarring with each other on different layers.

Therefore overlays should be set up with implicit or explicit awareness of the underlying network structure and protocols. Proposals for a better supply of the application layer with information about the network topology include measurement based estimation of Internet coordinates [14][24] and the installation of an information service or oracle collecting network layer information for support of higher layer protocols [1][6][23][32]. It remains a challenging design problem for future Internet architectures to keep the layering structure simple and at the same time to keep the flexibility to respond to new requirements and technical opportunities as the main motivation for the emergence of overlays of various types.

After a short discussion of the properties of P2P versus CDN content delivery, section 2 addresses the relevance of Zipf laws for access pattern and many other Internet characteristics. Section 3 compares currently discussed approaches for local content distribution followed by the conclusions.

2 Access to and Distribution of Content on the Internet via Overlays

Today, well-known and most popular peer-to-peer applications are file sharing and voice over IP (Skype) attracting millions of users, while a much wider spectrum of Internet application can more or less benefit from P2P networking. P2P solutions for online gaming, video streaming and IPTV also have a potential to extend to mass market and in addition, P2P overlay support is useful for small communities and enterprises.

Although P2P protocol designers have developed highly efficient distributed data transfer and control schemes, there are still unsolved open issues for all parties involved in P2P communication, including security concerns and uncontrolled distribution of partly illegal content in self-organizing communities. Business solutions based on P2P with controlled access and digital rights management have also been set up. The BBC iPlayer is such a publicly accessible P2P service for downloading television and radio programs of the last seven days, as a step towards integration of Internet and television [4].

On the other hand, popular client-server based web sites usually are supported by content delivery networks (CDN) [8]. A study of transfer paths in Akamai´s CDN [30] shows how users are redirected from the original web site by the underlying CDN to a close-by server within a hierarchical server farm consisting of thousands of servers. The connection is dynamically switched to another server if performance measurement and load conditions indicate better quality of service for the new path. Su et al. [30] confirm that CDNs are efficient in shortening transmission paths and improving delays and throughput as main quality of service characteristics. Similar experience has been made for Limelight as another CDN provider, although a comparative study [22] reveals that the Limelight server platform is hosted on an essentially smaller number of different locations.

2.1 CDN versus P2P Content Distribution

As compared to CDN transfers from a nearby server, P2P downloads usually experience much longer paths and delays which also affect throughput and reliability. For network providers, unnecessary long transfer paths impose higher load on peering and interconnection routes including expensive intercontinental links [6][23]. Figure 1 illustrates the different behaviour of CDN and P2P networks, depicting a usual topology of broadband access networks, where tree-shaped access regions are attached to edge routers of the backbone at points of presence (PoPs).

Considering large provider networks serving millions of subscribers, it can be expected that a majority of the data of a global file sharing network is already found to be replicated on the platform of the same Internet service provider (ISP) and often already in the same access region of a P2P downloader. The fact that the major portion of downloads is addressed to a small set of the currently most popular files strengthens this effect.

We have evaluated the delays for traffic via P2P and CDN overlays through packet based measurement on a link in the aggregation of Deutsche Telekom's broadband access network. We did not capture all traffic of both types, but selected a fraction that can be easily detected via P2P ports for BitTorrent, eDonkey and Gnutella and via IP address ranges indicating autonomous systems of Google, Akamai and Limelight server sites. The flows classified via P2P ports made only a fraction of 2.7% of the traffic volume, while the IP address ranges for CDNs and popular web sites accounted for 10.7% of the total traffic with a mean rate of about 820 Mb/s. The measurement was done for one hour at the daily peak rate in mid 2008. The measurement involved transmission in only one direction. We used two successive packets sent by the client in the TCP handshake to estimate the round trip delay, although this may also include reaction times of the server or peer in response to the TCP connection request.

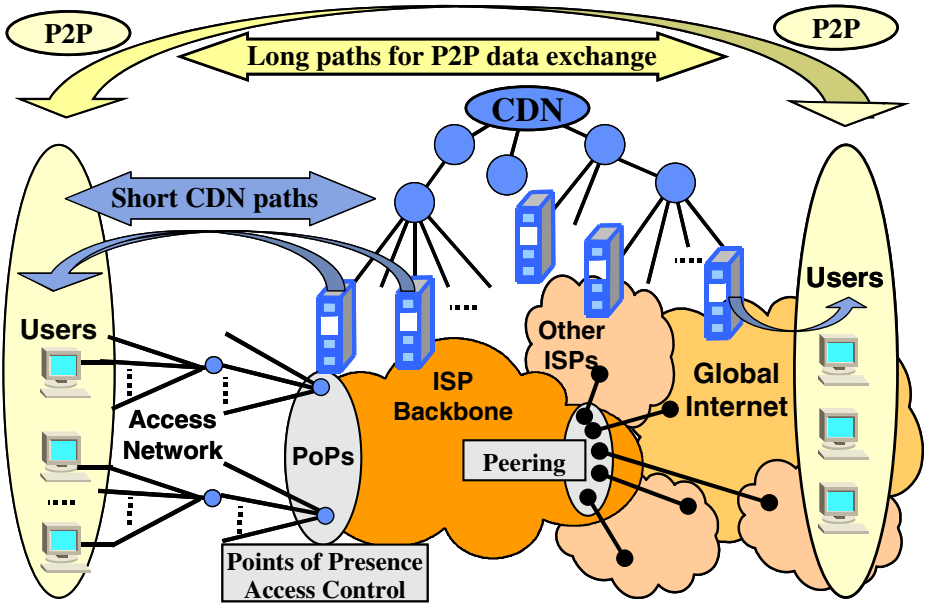


Fig. 1. Transport paths in P2P and CDN overlays

Figure 2 compares the cumulative distribution function (CDF) of those delays up to 1s as well as the fraction of delays falling into each 0.01s time slot for both the P2P and CDN flows. As expected, the delays are shorter for CDN delivery. The mean delay for CDN flows is 0.125s, whereas P2P flows have a mean delay of 0.33s with about 10% of the delays exceeding 1s.

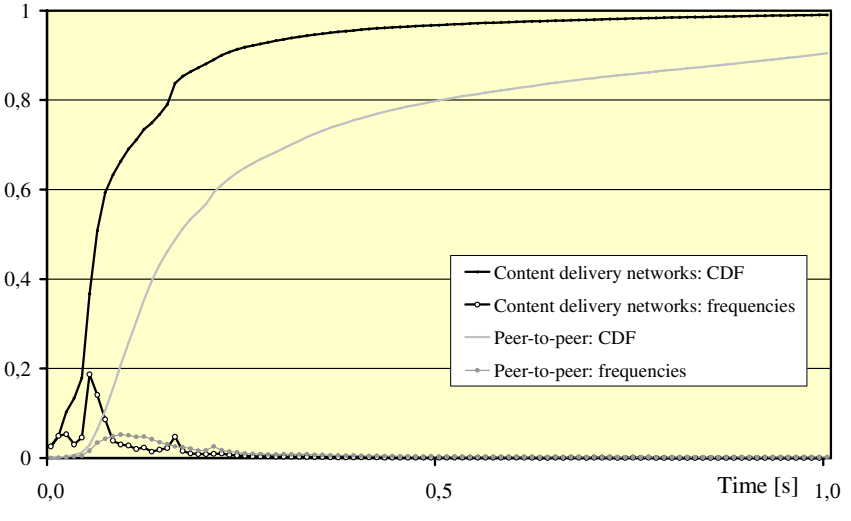


Fig. 2. Delay measurement for P2P and CDN overlays

In contrast to the generally behaviour visible in the measurement, a tendency for local exchange of P2P content can arise within social groups. A separation of user communities and content due to different languages is most obvious. When looking at downloads of German content to a German destination via eDonkey, it is not surprising to find an 80 : 20 rule, such that about 80% of the sources are again located in Germany, whereas the opposite, i.e. less than 20% of source locations are observed in Germany for downloads of English content [28]. The assignment of peers to the same supernode in the eDonkey network may also generate locality within the reach of a supernode, but which does not correspond to national or ISP boundaries.

Traffic locality also has been investigated for data exchange between regions within an ISP platform by Cho et al. [10] using measurement from several Japanese service providers. Based on IP address analysis, they found user-to-user traffic to be dominant, which may correspond to P2P or other applications running between users. The traffic matrices between regions are determined and do not reveal significant local dependencies, i.e. traffic is exchanged between neighboring regions at about the same rates as between distant regions.

The lack of regional locality in traffic matrices is also visible for Internet traffic in Germany, in contrast to traffic on voice platforms, e.g. ISDN, with a considerable portion of local calls. But even when voice networks migrate to voice over IP, voice will contribute only a small fraction of the total IP traffic and thus locality in IP traffic matrices is expected to stay at a low level.

2.2 The Relevance of Zipf Laws/Pareto Distributions in Internet Modelling

Pareto or equivalent Zipf distributions are observed manifold in Internet statistics or more general, when a large population is accessing a large set of items. Originally V. Pareto introduced the form to characterize the distribution of property and income over the population, including the effect of extreme outliers, such that a small fraction of the largest items still has essential effect on the mean.

According to a Zipf law, the item that has rank R in the order of highest access frequency attracts

$$A(R) = \alpha R^{-\beta} \qquad (\alpha > 0; 0 < \beta < 1) \tag{1}$$

accesses. In the same form of equation (1), $A(R)$ can express the portion of accesses addressed to the item, when α is determined according to a normalization constraint, such that $\Sigma_R A(R) = 1$. The items may refer to videos available on an Internet portal, e.g. via the BBC iPlayer, America Free TV or YouTube, or book, DVD sellings etc. The hit rate for a fraction of the most popular items in Zipf law distributed accesses is determined by the sum of access frequencies for the first N items:

$$\sum_{R=1}^{N} R^{-\beta} > \int_{R=1}^{N+1} R^{-\beta} dR = \frac{R^{1-\beta}}{1-\beta}\Bigg|_1^{N+1} = \frac{(N+1)^{1-\beta} - 1}{1-\beta} \tag{2}$$

since the sum is equivalent to the integral of a step function $f_U(R) = \lfloor R \rfloor^{-\beta} \geq R^{-\beta}$ as an upper bound of the real valued function $R^{-\beta}$. Vice versa, we obtain the lower bound

$$f_L(R) = \lfloor R+1 \rfloor^{-\beta} < R^{-\beta} \Rightarrow$$
$$\sum_{R=1}^{N} R^{-\beta} = 1 + \sum_{R=1}^{N-1} (R+1)^{-\beta} < 1 + \int_{R=2}^{N+1} R^{-\beta} dR < \frac{(N+1)^{1-\beta} - \beta}{1-\beta}$$

The bounds on the sum can be used to determine the factor α in $A(R) = \alpha R^{-\beta}$ according to the total number of accesses $\sigma_A = \Sigma_R A(R)$:

$$\frac{(1-\beta)\sigma_A}{(N+1)^{1-\beta} - \beta} < \alpha < \frac{(1-\beta)\sigma_A}{(N+1)^{1-\beta} - 1}$$

For $\sigma_A = 1$, $A(R)$ corresponds to the portion of accesses. As an alternative, the fact that $A(1) = \alpha$ can be used to make α fit to the tail of the most often accessed items of the distribution.

Zipf-like distributions for web sites have been investigated by Breslau et al. [7] based on a set of measurements from half a dozen different networks, yielding parameters in the range $0.64 < \beta < 0.85$. Recent studies of access to YouTube [9][16] again show Zipf-like distributions although with deviation for seldom accessed videos. M. Eubanks [15] reports similar results for the popularity of content on America Free TV <americafree.tv> and for other cases. Thus, an essential portion of P2P data transfers could already be found and delivered within an ISP's platform and even without crossing the backbone.

The typical effect of a Zipf law concentrates a large portion of all accesses on a few most popular items. As a consequence, caches can be efficient by storing a small set of those items locally to shorten the access paths to the users. But it is apparent from equation (2) that the Zipf law effect does not hold for arbitrary large sets of items, since summation over the function

$$\lim_{N\to\infty} \sum_{R=1}^{N} R^{-\beta} > \lim_{N\to\infty} \frac{(N+1)^{1-\beta} - 1}{1-\beta} \to \infty$$

does not converge and the portion of accesses to the most popular top K items is decreasing to zero when the population N becomes very large. Therefore Zipf laws are usually observed only for a limited range of popular items [7][15][16], while the access distribution deviates in the region of less popular ones.

As an example, Figure 3 compares the statistics of the traffic volume for user sessions to a Zipf law. The volume statistics is based on more than 4500 sessions taken from measurement in the aggregation platform of Deutsche Telekom's broadband access network over two afternoon hours in mid 2008.

An upper curve shows the cumulative distribution function (CDF) of the number of sessions with regard to their volume. A second CDF curve is for the portion of the total volume that is contributed by those sessions in increasing order. In comparison, both curves confirm a 90:10 rule, such that only 10% of the sessions contribute about 90% of the total traffic, where each of those largest sessions generates a traffic volume of more than 10MB during the measurement. On the other hand, 71% of the sessions generated volumes below 1MB each and together less than 1% of the total volume.

A third CDF curve has been adapted according to a Zipf law for the session volume size $V(R) = 0.027 \cdot R^{-0.8}$, where $\beta = 0.8$ is determined to match the slope of the lower curve and $\alpha = V(1) = 2.7\%$ is the portion that the largest session contributes to the total volume. This example shows an almost perfect match for the largest sessions, although again severe deviations are visible for small sessions up to the 20MB range.

This result on the asymmetry of traffic volumes generated per session or per access raises the question about the fairness of flat rate accounting. Although the trend to broader multi purpose usage detracts from the dominance of P2P traffic, ISPs and users still face the problem of inadequate charging for the majority of traffic on the Internet [29]. In Germany and Europe, subscribers prefer simple flat rate access without accounting for the traffic volume, which is offered with only minor differentiation in access speeds. Volume based tariffs combined with a low monthly fee and flat rate up to a volume limitation would be sufficient for most users, but then they are often left without awareness or control of costs in excess of the limit until a next monthly bill. Even if the costs for transport of high traffic volumes in the backbone is only a part of the access provisioning costs, flat rate pricing seems to privilege the heavy traffic producers on account of all other users.

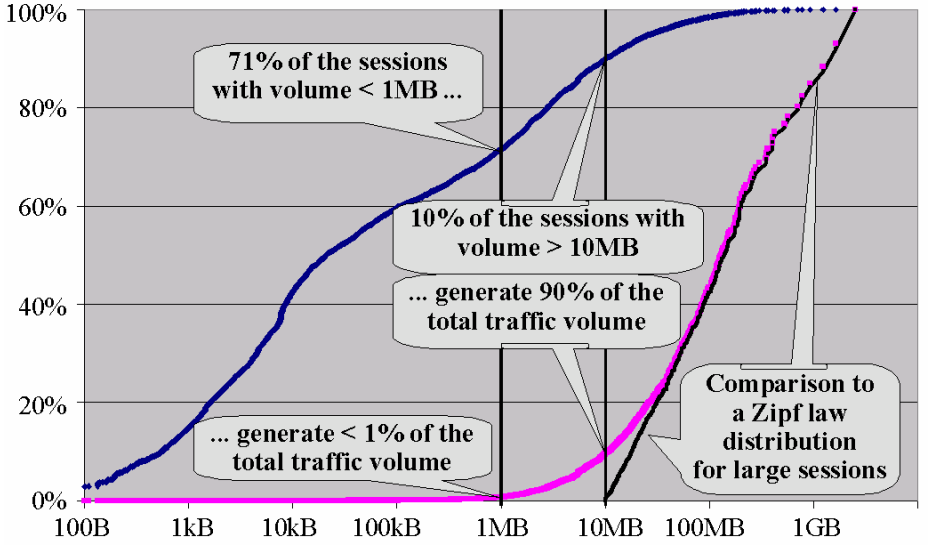


Fig. 3. CDF of session volumes compared to a Zipf law. Upper curve: Fraction of the number of sessions with volume < x Byte. Lower curve: Fraction of total volume contributed by sessions with volume < x Byte.

As final remarks on the relevance of Zipf or Pareto distributions in Internet modelling, we refer to the fact that transmission of Pareto distributed transfer volumes with a memoryless Poisson arrival process causes self-similar traffic pattern, which is often observed in Internet traffic measurement, although not always in pure form [19]. The interconnection structure of the Internet also exhibits a Zipf law regarding the number of connections between web sites or between autonomous systems. Albert and Barabasi [2][3] have confirmed that many of the largest social networks show

similar properties of scale free networks. Scale free networks can be built randomly by inserting new nodes such that the probability of connecting a new node to an existing network node is proportional to the current degree of the node [18]. In this way, node degrees approach a Pareto distribution when the network becomes large, while the diameter of the network stays small (*small world effect*) and is estimated at about 20 for the current Internet connection structure [3][18].

3 Optimizing Data Access Paths

The problem of replacing long transmission paths between peers by local data exchange among peers close to each other is addressed in a number of recent approaches. Information servers are proposed as depicted in Figure 4 [1][6][31][32], which can be queried by applications in order to localize data sources in the near based on information from network providers and other parties. Alternatives are coordinate systems to be established for the sources of distributed applications [14][24] and caches [27], whose efficiency profits from the relevance of Zipf laws in access pattern. With knowledge of the position of peers or servers, the data flows of an application can be optimized, even regarding bottlenecks on transmission links [32]. Since some of those proposals have to rely on a common and standardized concept, the Internet Engineering Task Force (IETF) has started activity in this area.

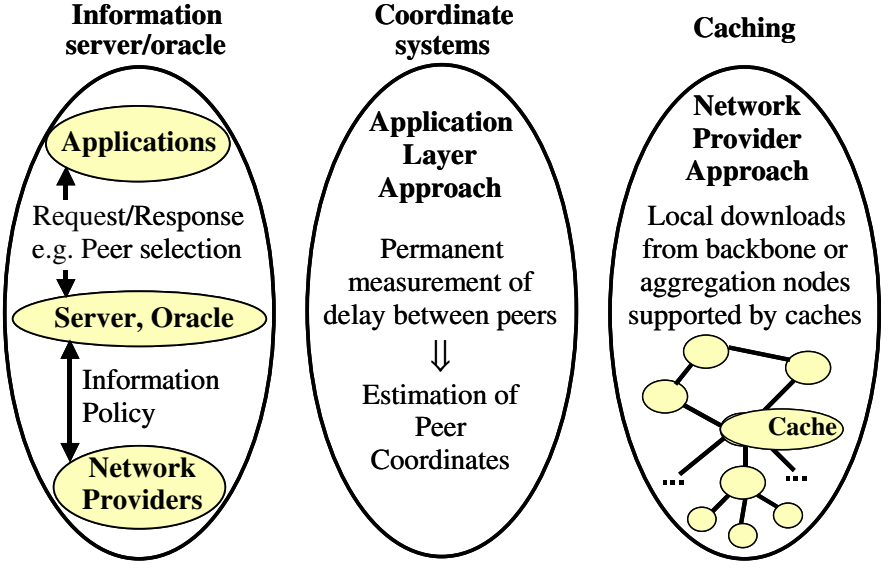


Fig. 4. Approaches for localized content delivery

3.1 Current Ietf Discussion on Application Layer Traffic Engineering

In 2008, the Internet Engineering Task Force (IETF) has set up a working group on application layer traffic optimization (ALTO) [23], aiming at a standardized support to improve localized data exchange in P2P and other types of content distribution networks.

The basic problem to be addressed is how distributed applications can get information about the locations and distances between the involved hosts or peers. A main focus is on cooperation between network operators and other parties who are able to provide location information for application protocols. Several projects and studies are already investigating approaches for a globally accessible information service that can be fed by parties with knowledge about network topologies in the Internet [1][6][31][32]. Distributed applications can access the information server through queries in order to optimize host or peer relationship for data exchange. As a starting point, information services about locations are already online, e.g. the Prefix WhoIs service <www.pwhois.org> or <www.closestnode.com>. Main preconditions and aspects of a standardized extension of a location information service are

- Which kind of metrics and information is useful to estimate distances and local relationship between different hosts?
- Which parties are expected to provide such information based on which own interest or incentives?
- How should an interface to the service be designed for access by applications?
- How can availability and scalability be achieved and failure cases be handled?
- What is the effect of the location information service for users and traffic pattern in network provider platforms? How can it be measured and optimized?

Two basic methods are considered for traffic path optimization:

- IP address mapping as done by the Prefix WhoIs service, for assigning the autonomous system to an IP address as a crude classification or at a finer granularity and
- Probing methods to measure delay, throughput or other performance parameters. This approach is most sensitive and responsive to current network conditions, but requires considerable effort and overhead. It may be too time consuming for small data transfers. Probing can be implemented on the application layer in P2P protocol without involving servers.

A coordinate system built on probing for delays between BitTorrent peers has been studied and implemented in prototype versions of the Azureus client [24]. The probing is piggybacked on other messages between peers to reduce probe messaging overhead. As a result, a two dimensional coordinate system is constructed with an additional height component to account for delays in the access. Much effort is needed to maintain a coordinate system. The variability of delay measurement results over time, changes in the routing and churn are among the main factors detracting from the accuracy. The study [24] concludes that useful coordinates can be established on application layer, but the effort to employ a coordinate system seems to be affordable only for large scale P2P networks.

A server system can combine information gathered from several sources to obtain more precise information with less effort. When a P2P download is initiated, the P2P protocol usually determines a list of possible sources offering the required content, which is handed over to the client, who connects to several proposed sources until a sufficient throughput is obtained in a multi source download process. While popular P2P protocols currently choose the sources more or less at random, a server with topology awareness can rank sources in the list due to distances from the requesting

peer. In the ranked list, the client can start choosing close-by sources, which may also be preferable with regard to throughput and delay, when they still have enough upstream capacity available. The solution again involves some overhead and has to be included into each P2P protocol, depending on servers to collect information from network providers or other parties who contribute knowledge on Internet topology and distances.

3.2 Experience from the P4P Project Including Traffic Engineering

A testbed for application layer traffic optimization has already been set up by the P4P project <codex.cs.yale.edu/avi/home-page/p4p-dir/p4p.html> and a P4P working group hosted by the Distributed Computing Industry Association <www.dcia.info>. Recent results of a field trial [32] claim that the mean path length for P2P downloads could be reduced from 5.5 to 0.9 metro-hops within the Verizon intra-domain network. In general, the gain for intra-domain paths depends on the size and the structure of a provider platform, which often spans a smaller area with less than 5 hops to cross the backbone.

On the other hand, more improvement is expected for inter-domain traffic paths, which traverse the boundaries of network platforms partly on expensive intercontinental links. Network providers would profit a lot from redirecting traffic paths into their platform wherever possible. A classification of sources according to autonomous systems (AS), which usually correspond to network regions under common administration, is helpful to support this approach. Since not all P2P traffic can be bound within an AS, network providers could add policy information about which neighboring ASes are preferable for connections to peers in their platform, based on their knowledge of inter-AS connectivity and bottlenecks as well as the expenses for utilizing inter-AS links.

A corresponding inter-domain scenario has also been investigated by the P4P project [32], considering usual BitTorrent traffic pattern and volume according to measurement in the Abilene network in 2007. In one scenario, the traffic on the assumed inter-AS links is reduced to 2/3 by preferring sources based on low delay, whereas a second server-based scenario including the P4P traffic optimization method even yields less than 40% of the original P2P traffic load on inter-AS links.

The application layer traffic engineering approaches studied in the P4P project include load balancing to avoid bottlenecks and to minimize costs associated with traffic on network links. Nevertheless, this raises the question whether the network providers or the applications should be responsible for traffic engineering in the future Internet and what is the overall effect when this is done on the network and on the application layer? Counterproductive scenarios may arise, e.g. when the application layer optimization reorganizes the overlay network structure in order to avoid a bottleneck link. As a consequence, the traffic matrix on the network layer will observe a corresponding shift in traffic demands and the network provider may no longer see bottlenecks and the need to upgrade them, as illustrated in Figure 5.

In practice, traffic engineering has to take link failures into account as well as a link upgrading process for steadily increasing Internet traffic [25]. On the network layer, load balancing solutions are available for that purpose [13][20], but it seems challenging to combine or even integrate them with similar application layer functions under different administration.

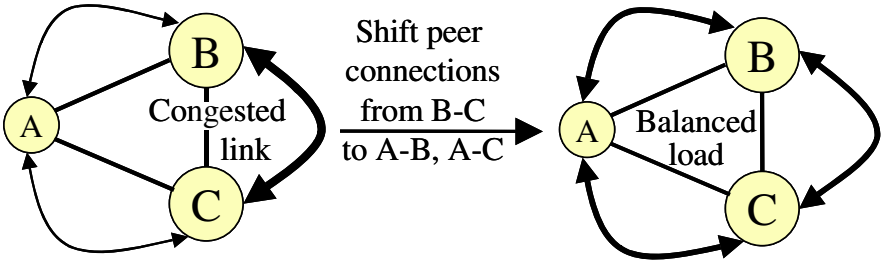


Fig. 5. Load balancing on the network and application layer?

3.3 Influence of Biased Source Selection on the Overlay Topology

The proposed biased source selection for exploiting local data exchange has an impact on the overlay network topology. Random source selection leads to a strongly connected global mesh network without hierarchical or regional structures. Randomly built graphs develop towards scale free networks with small world effects. The global Internet structure is based on random and scale-free properties [2][3][18].

Biased local source selection strengthens a tendency to build a number of clusters with strong internal connectivity and a loose coupling between those clusters. With preference for connections between nodes in the same AS, clusters or subnet structures can be expected within each AS. Thus improved locality may slow down the distribution of content over network boundaries of ISPs. In the worst case, strict local preference and a high churn in P2P networks may lead to separated subnets for the same content.

Thus, sufficient inter-domain connectivity has to be sustained. An obvious approach is to mix biased local preference with randomness. Then the portion of random selection has to keep the inter-domain traffic throughput at a smallest possible level in a trade-off between avoiding a slow down of content propagation around the globe. The portion depends on the size of the overlay and can be smaller in large overlays, allowing for more efficient localization. Alternative approaches to maintain a structured and optimized overlay network for inter-domain connections have to cope with the randomness and churn in P2P networks.

3.4 Caches

Web caches provide another opportunity to optimize traffic paths [17][27]. In principle, caches are highly efficient for P2P data, again because of Zipf-like distributions [7][15][16] for accesses to content. Therefore storing a small fraction of most popular content is sufficient to attract a high hit rate. Problems of outdated data as in classical web caches are less relevant for most high volume video content or for uniquely identified P2P data using hash algorithms. Caches can be placed in aggregation nodes for broadband access close to the subscriber lines yielding minimal transport path lengths.

On the other hand, the problem of supporting the distribution of copyright infringing or other illegal content by caches is apparent. In principle, the content of web caches always reflected content of the Internet, including problematic content although in an

agnostic way. Nevertheless, the problem now has attracted much more attention and is pursued by various counteractions since file sharing is suspected to be responsible for decreasing revenues for content providers in the music and film industry. Filtering or classification of content in a cache is as difficult as filtering web content requiring expensive deep packet inspection methods. Since users distrust content inspection which can be seen in conflict with privacy laws of many countries, ISPs should use filtering only on behalf of legal authorities rather than on their own policies.

Since October 2004, the eDonkey/eMule P2P network offered an option to use the web caches of network providers by disguising P2P downloads as usual HTTP web browsing requests. When investigating the usage of this cache option in 2006, only a small portion of 5-10% of the download volume was observed to be supported by caches, although cases of downloading from the cache achieved essentially, often 5-fold higher throughput [28]. Later protocol versions did not proceed with cache support, see <wiki.emule-web.de/index.php/webcache>.

While file sharers obviously could benefit from cache downloads at higher throughput, the alternative of biased source selection with local preference is more favourable for network providers. The entire throughput of a P2P overlay is limited by the upstream capacities of the participating peers as a bottleneck especially in ADSL access. For biased source selection, this bottleneck remains unchanged even if it may be better exploited. But when caches are included, then the access bandwidth of the caches becomes available for downloading in addition to the upstream bottleneck. Although the bandwidth of a cache is under control of the service provider, file sharing can utilize the cache as well as the P2P overlay to maximize their throughput. The network provider may experience P2P traffic load to persist in the backbone and even to increase in the access due to cache support. From the user perspective, caches in the access would be ideal to improve the throughput and to shorten transport paths and corresponding delays.

In principle, network providers can offer the most efficient support for content distribution using their own infrastructure including multicast/broadcast services. In this way, a single provider can't build architectures of global reach, but it is appropriate for offering regional services or IPTV within a country or lingual community.

3.5 Traffic Measurement and Control Options

Finally, network providers can try to enforce locality by controlling and reducing the traffic especially on expensive links on peering and transatlantic routes. For that purpose, P2P traffic has to be classified and differentiated from other traffic. There are a number of vendors of application identification systems and experience has been published for detection of main P2P protocols [6][21][26]. Reducing P2P traffic on the borders of an ISP network would again give preference for shorter intra-domain download paths.

Nevertheless, P2P traffic classification is subject to non-negligible effort, as demonstrated by more than 1000 behaviour patterns being included in a thorough decision process for classification of BitTorrent [21].

4 Conclusions

Frequently accessed content on the Internet has to be delivered on efficient short paths from nearby servers or distributed sources, where peer-to-peer networks currently leave a large potential open for optimization. We have investigated the alternatives to optimize the traffic paths via CDN and P2P overlays with support from location servers, caches or measurement based traffic engineering. The relevance of Zipf laws in access pattern is favourable for caching of popular content or other distributed schemes based on a large number of devices with limited storage.

In the recently started IETF standardization process it is still open which approaches will be implemented and will play an important role in the future Internet. Nevertheless, progress can be expected towards more efficient localized transport on network platforms from which the users can also benefit in terms of shorter delay and higher throughput. The efficiency of P2P as well as CDN approaches in avoiding unnecessary traffic load and minimizing delays on short paths is a decisive factor to stay competitive, where combined P2P/CDN solutions [22] are a promising option.

References

[1] Agrawal, V., Feldmann, A., Schneideler, C.: Can ISPs and P2P users cooperate for improved performance? ACM SIGCOMM Comp. Comm. Review 37, 31–40 (2007)

[2] Albert, R., Barabasi, A.-L.: Statistical Mechanics of Complex Networks. Rev. Mod. Phys. 74(47), 1–54 (2002)

[3] Barabási, A., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)

[4] BBC iPlayer (since 2007), http://www.bbc.co.uk/iplayer/

[5] Biersack, E., et al.: Overlay architectures for file distribution: Fundamental analysis for homogeneous and heterogeneous cases. Computer Networks 51, 901–917 (2007)

[6] Bindal, R., et al.: Improving traffic locality in BitTorrent via biased neighbor selection. In: ICDCS (2006)

[7] Breslau, L., et al.: Web caching and Zipf-like distributions: Evidence and implications. In: Proc. IEEE Infocom (1999)

[8] Canonico, R., Fuererro, C., Mauthe, A. (eds.): Content distribution infrastructures for community networks. Computer Networks Special Issue 53(4), 431–568 (2009)

[9] Cheng, X., Dale, C., Liu, J.: Statistics and social network of YouTube videos. In: IEEE Proc. Interna-tional Workshop on Quality of Service (IWQoS), Twente, The Netherlands, pp. 249–258 (2008)

[10] Cho, K., Fukuda, K., Esaki, H., Kato, A.: The impact and implications of the growth in residential user-to-user traffic. In: ACM Sigcomm Conf., Pisa (2006), http://www.acm.org/sigs/sigcomm/sigcomm2006

[11] Cisco Systems, Global IP traffic forecast and methodology, White paper (2008), http://www.cisco.com

[12] Cohen, B.: Incentives build robustness in BitTorrent (2003), http://www.bitconjurer.org/BitTorrent/bittorrentecon.pdf

[13] Cohen, R., Nakibly, G.: Maximizing restorable throughput in MPLS networks. In: Proc. IEEE INFOCOM mini-conference, Phoenix, USA (2008)

[14] Dabek, F., Cox, R., Kaashoek, F., Morris, R.: Vivaldi: A decentralized network coordinate system. In: ACM Sigcomm Conf., Portland, USA (2004), http://www.conferences.sigcomm.org/sigcomm/2004/

[15] Eubanks, M.: The video tsunami: Internet television, IPTV and the coming wave of video on the Inter¬net, Pleanry talk, 71. IETF meeting (2008), http://www.ietf.org/proceedings/08mar/slides/plenaryt-3.pdf

[16] Gill, P.: YouTube workload characterization, Master Thesis, Univ. of Calgary, Canada (2008), http://www.pages.cpsc.ucalgary.ca/~psessini/papers/pgill_thesis.pdf

[17] Hasslinger, G.: ISP Platforms under a heavy peer-to-peer workload. In: Steinmetz, R., Wehrle, K. (eds.) Peer-to-Peer Systems and Applications. LNCS, vol. 3485, pp. 369–381. Springer, Heidelberg (2005)

[18] Hasslinger, G., Kempken, S.: Applying random walks in structured and self-organizing networks: Evaluation by transient analysis. In: PIK – Special Issue on Self-Organizing Networks, vol. 31, pp. 17–23. Saur Verlag (2008)

[19] Hasslinger, G., Mende, J., Geib, R., Beckhaus, T., Hartleb, F.: Measurement and characteristics of traffic in broadband access networks. In: Mason, L.G., Drwiega, T., Yan, J. (eds.) ITC 2007. LNCS, vol. 4516, pp. 998–1010. Springer, Heidelberg (2007)

[20] Hasslinger, G., Schnitter, S., Franzke, M.: The Efficiency of Traffic Engineering with Regard to Failure Resilience. Telecommunication Systems 29(2), 109–130 (2005)

[21] Hu, Y., Chiu, D., Lui, J.: Application identification based on network behavioral profiles. IEEE Proc. 16. Workshop on Quality of Service (IWQoS 2008), Twente/Enschede, The Nederlands, pp. 239–248 (2008)

[22] Huang, C., Wang, A., Li, J., Ross, K.: Understanding Hybrid CDN-P2P. In: Proc. NOSSDAV Conf., Braunschweig, Germany, pp. 75–80 (2008)

[23] Internet Engineering Task Force (IETF), ALTO working group: Application layer traffic optimization, http://www.ietf.org/html.charters/alto-charter.html

[24] Ledlie, J., Gardner, P., Seltzer, M.: Network coordinates in the wild. In: Proc. USENIX Conf., pp. 299–311 (2007)

[25] Odlyzko, A.: Internet traffic growth: Sources and implications. In: Proc. SPIE, vol. 5247, pp. 1–15 (2003), The Minnesota Internet traffic studies (MINTS), http://www.dtc.umn.edu/mints/references.html (continuously updated until 2008)

[26] Rossenhövel, C.: P2P filters ready for Internet prime time? http://www.internetevolution.com/document.asp?doc_id=148803

[27] Saleh, O., Hefeeda, M.: Modeling and caching of P2P traffic. In: Proc. IEEE Internat. Conf. on Network Protocols (2006)

[28] Schroeder-Bernhardi, J.: Analysis of the communication and traffic in P2P networks including web caches, Master Thesis, KOM-D-260, Univ. of Darmstadt, Germany (2006)

[29] Sigurdsson, H.M., Halldorsson, U.R., Hasslinger, G.: Potentials and Challenges of Peer-to-Peer Based Content Distribution. Telematics and Informatics, vol. 24, pp. 348–365. Elsevier, Amsterdam (2007)

[30] Su, A.-J., Choffnes, D.R., Kuzmanovic, A., Bustamante, F.E.: Drafting behind Akamai. In: Proc. ACM SIGCOMM, Pisa, Italy (2006)

[31] Walkowiak, K.: A Flow Deviation Algorithm for Joint Optimization of Unicast and Anycast Flows in Connection-Oriented Networks. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008, Part II. LNCS, vol. 5073, pp. 797–807. Springer, Heidelberg (2008)

[32] Xie, H., et al.: P4P: Provider Portal for Applications. In: Proc. ACM SIGCOMM, Seattle, USA (2008)

Solving Multiserver Systems with Two Retrial Orbits Using Value Extrapolation: A Comparative Perspective

M[a] Jose Domenech-Benlloch[1], Jose Manuel Gimenez-Guzman[2], Vicent Pla[1], Vicente Casares-Giner[1], and Jorge Martinez-Bauset[1]

[1] Dept. Comunicaciones, Universidad Politecnica Valencia
Cami de Vera, s/n 46022, València, Spain
[2] Dept. Automatica, Universidad de Alcalá
28871 Alcalá de Henares, Madrid, Spain
mdoben@doctor.upv.es, josem.gimenez@uah.es,
{vpla,vcasares}@dcom.upv.es, jmartinez@upvnet.upv.es

Abstract. In communication networks that guarantee seamless mobility of users across service areas, reattempts occur as a result of user behavior but also as automatic retries of blocked handovers. A multiserver system with two reattempt orbits is obtained when modeling these networks. However, an exact Markovian model analysis of such systems has proven to be infeasible and resorting to approximate methods is mandatory. To the best of our knowledge all the existing methods are based on computing the steady-state probabilities. We propose another approach based on the relative state values that appear in the Howard equations. We compare the proposed method with the most well-known methods appeared in the literature in a wide range of scenarios. The results of the numerical evaluation carried out show that this solution outperforms the previous approaches in terms of both accuracy and computation cost for the most common performance parameters used in retrial systems.

Keywords: Wireless and Mobile Systems and Networks (WLAN, 2G-3G-4G), Queueing Systems and Networks, Stochastic Models, Markov Models, Performance Modelling.

1 Introduction

The retrial phenomenon appears in multiple situations in telecommunications and computer networking. In this paper, we focus our attention on a generic communication network that guarantees seamless mobility to its customers by means of a cellular architecture. In this type of networks, the network coverage area is divided into cells and customers can move across different cells of the network. When a customer with an active communication moves from one cell to another, a so-called handover procedure is executed. Nowadays, perhaps the most widespread and popular example of this type of networks are the cellular telephone networks —2G and 3G— but the current perspective is that in near

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 56–70, 2009.

future a variety of technologies fitting into this category will be in place, e.g., Mobile IP, IEEE 802.16e —WiMAX— and IEEE 802.20 —Mobile Broadband Wireless Access, MBWA.

This paper deals with the case in which reattempts appear not only when a customer is blocked but also when a handover is blocked as in GSM [1]. To the best of our knowledge, the first and only paper that has considered the effect on network performance of both types of reattempts simultaneously is [2]. Now, in this paper, we refer to the former as redials and to the latter as (automatic) retrials, while we use the term reattempt to refer to any of them. Blocked handovers will be automatically retried until a reattempt succeeds or the user moves outside the handover area. In the former case the session will continue without the user noticing any disruption, while in the latter the session will be abruptly terminated. In contrast, persistence of redials depends on the user patience and an eventual abandonment results in session setup failure. Another difference is that the maximum number of unsuccessful automatic retrials is set by the network operator while redials are affected by the randomness of human behavior. Therefore, both types of reattempts have different characteristics and as a consequence two separate retrial orbits have to be considered in the analysis of the system.

The modeling of repeated attempts has been the subject of numerous investigations. Two functional blocks are typically distinguished in models which consider reattempts: a block that accommodates the servers and possibly a waiting queue, and a block where users that reattempt are accommodated, usually called reattempt orbit. It is known [3] that to solve this type of systems it is necessary to resort to approximate methods. These methods are usually grouped into three categories: approximations, finite truncated methods and generalized truncated methods [3,4]. We will direct our attention only to finite and generalized truncated methods. The finite truncated methods replace the original infinite state space by a finite one, where steady-state probabilities can be computed. On the other hand, generalized truncated methods replace the original infinite state space by another infinite but solvable state space. This last type of methods usually outperforms the other two types [4].

All the approaches presented in the literature so far rely on the numerical solution of the steady-state Kolmogorov equations of the *Continuous Time Markov Chain* (CTMC) that describes the system under consideration. Very recently, however, an alternative approach for evaluating infinite state space Markov processes has been introduced by Leino et al. [5]. The new method, named Value Extrapolation (VE), does not rely on solving the global balance equations, but considers the system in its Markov Decision Process (MDP) setting and solves the expected value from the Howard equations written for a truncated state space.

The main objective of this work is to tailor the VE method to a system with two reattempt orbits and compare its performance with the performance of other possible approximate methods. This performance evaluation is done in a cellular network scenario that guarantees seamless mobility to its users. We conclude

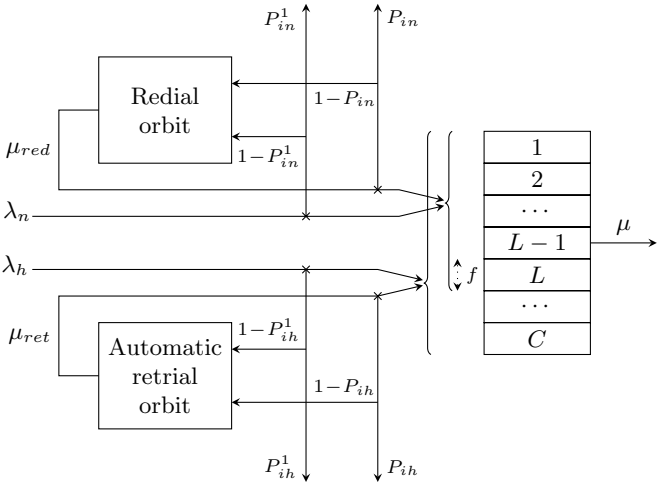


Fig. 1. System model

that VE greatly outperforms the rest of the methods throughout a wide range of scenarios not only in terms of accuracy, but also in terms of computation cost, so its use is highly recommendable.

The rest of the paper is structured as follows. First, we describe the cellular network under study and its associated model. In Section 3, we enumerate and explain the main features of the methods we compare VE with. Section 4 is devoted to the description of VE and how it has been applied to the model under study. A numerical study is performed in Section 5 and finally, a summary of the paper and some concluding remarks are given in Section 6.

2 System Description and Model

We consider a cellular mobile network with a fixed channel allocation scheme and where each cell is served by a different base station, where C is the number of resources in the cell. As shown in Fig. 1 there are two arrival streams: the first one represents new sessions and the second one handovers from adjacent cells. Both arrival processes are considered to be Poisson with rates λ_n and λ_h respectively. This leads to an overall arrival rate of $\lambda = \lambda_n + \lambda_h$. For the sake of mathematical tractability, the channel holding time is assumed to be exponentially distributed with rate μ [6].

In general, blocking a new session setup is considered to be less harmful than blocking a handover attempt. Therefore, we must include an admission control policy to guarantee the prioritization of handovers —and retrials— over new sessions —and their associated redials— and therefore, assure a certain degree of Quality of Service (QoS). The most widespread technique is to reserve some resources to highest priority flows, being in our case handovers and their associated

automatic retrials. This technique can be generalized to a fractional reservation, the so-called Fractional Guard Channel (FGC) admission control policy [7]. The FGC policy is characterized by only one parameter t $(0 \leq t \leq C)$. New sessions and redials are accepted with probability 1 when there are less than $L = \lfloor t \rfloor$ resources being used and with probability $f = t - L$, when there are exactly L resources in use. If there are more than L busy resources, new sessions and redials are no longer accepted. Handovers and automatic retrials are only rejected when the system is completely occupied.

When an incoming new session is blocked, according to Fig. 1, it joins the redial orbit with probability $(1 - P_{in}^1)$ or leaves the system with probability P_{in}^1. If a redial is not successful, the session returns to the redial orbit with probability $(1-P_{in})$, redialing after an exponentially distributed time with rate μ_{red}. Redials are able to access to the same resources as the new sessions. Note that P_{in}^1 and P_{in} model the impatience phenomenon of leaving the system without having been served. Similarly, P_{ih}^1, P_{ih} and μ_{ret} are the analogous parameters for automatic retrials. There are several performance parameters that are generally used to describe the behavior of this type of cellular systems with retrials and redials. On the one hand, the widely used blocking probabilities for both new sessions (P_b^n) and handovers (P_b^h). On the other hand, the mean number of users redialing (N_{red}) and handovers retrying (N_{ret}) can describe more accurately the reattempt phenomenon.

The model considered can be represented as a tridimensional (k, m, o) CTMC, where k denotes the number of sessions being served, m specifies the number of sessions in the redial orbit and o represents the number of sessions in the retrial orbit. The state space can be represented by:

$$\mathcal{S} := \{(k, m, o) : k \leq C; m \in \mathbb{Z}_+; o \in \mathbb{Z}_+\}.$$

The transition rates of this model are represented in Table 1. The main mathematical features of this queueing model consist of having two infinite dimensions —the state space of the model is $\{0, \ldots, C\} \times \mathbb{Z}_+ \times \mathbb{Z}_+$— and the space-heterogeneity along them. This heterogeneity is produced by the retrial and redial rates, which respectively depend on the number of customers in the retrial and the redial orbits.

3 Solving Methods

It is known that the classical theory, see, e.g., [8], is developed for random walks on the semi-strip $\{0, \ldots, C\} \times \mathbb{Z}_+$ with infinitesimal transitions subject to conditions of space-homogeneity. When the space-homogeneity condition does not hold the problem of calculating the equilibrium distribution has not been addressed beyond approximate methods [9]. Indeed, if we focus on the simpler case of multiserver retrial queues with only one retrial orbit, the absence of closed form solutions for the main performance characteristics when $C > 2$ can be enphasized [3].

Table 1. Transition rates of the exact model

Transition	Condition	Rate
$(k,m,o) \rightarrow (k+1,m,o)$	$0 \leq k \leq L-1$	λ
	$k = L$	$\lambda_h + f\lambda_n$
	$L < k < C$	λ_h
$(k,m,o) \rightarrow (k+1,m,o-1)$	$0 \leq k \leq C-1$	$o\mu_{ret}$
$(k,m,o) \rightarrow (k,m,o-1)$	$k = C$	$o\mu_{ret}P_{ih}$
$(k,m,o) \rightarrow (k+1,m-1,o)$	$0 \leq k \leq L-1$	$m\mu_{red}$
	$k = L$	$m\mu_{red}f$
$(k,m,o) \rightarrow (k,m-1,o)$	$k = L$	$m\mu_{red}(1-f)P_{in}$
	$L < k \leq C$	$m\mu_{red}P_{in}$
$(k,m,o) \rightarrow (k-1,m,o)$	$1 \leq k \leq C$	$k\mu$
$(k,m,o) \rightarrow (k,m,o+1)$	$k = C$	$\lambda_h(1-P_{ih}^1)$
$(k,m,o) \rightarrow (k,m+1,o)$	$k = L$	$\lambda_n(1-P_{in}^1)(1-f)$
	$L < k \leq C$	$\lambda_n(1-P_{in}^1)$

Obviously, to solve the system under study, it will also be necessary to resort to approximate models and numerical methods of solution. Although other approaches exist, for the comparison against VE we have chosen the three most well-known methods that are able to solve the problem under study. These methods are explained in the next subsections.

3.1 Double Truncation (DT)

The easiest and more intuitive method to solve the proposed model lies in the truncation of the infinite dimensions of the state space [10]. In our case, it must be applied to both the redial and retrial orbits, truncating them beyond levels Q_n and Q_h respectively and obtaining the state space:

$$\mathcal{S} := \{(k,m,o) : k \leq C; m \leq Q_n; o \leq Q_h\}.$$

Obviously, by increasing the values of Q_n and/or Q_h the considered state space in the approximation is enlarged and the accuracy of the solution is expected to improve at the expense of a higher computational cost.

The stationary probability distribution can be obtained by solving $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$ along with the normalization condition. As $\mathbf{Q}$ is a finite matrix this system can be solved by any of the standard methods defined in classical linear algebra [11].

3.2 Double FM (DFM)

As DT, DFM belongs to the family of finite truncated methods [3]. These methods consist of replacing the original infinite state space by a finite one. However,

DFM is more sophisticated than DT as it introduces in some sense the effect of the truncated states.

In [12] we developed FM, a generalization of the approximation method proposed in [13]. Although developed initially for a single orbit scenario, FM was applied to a system like the one under study in [14]. In this case FM has been applied to both retrial and redial orbits —resulting in DFM—, reducing the state space to a finite set by aggregating all states beyond a given occupancy of the orbits, producing the same approximate state space as DT:

$$\mathcal{S} := \{(k, m, o) : k \leq C; m \leq Q_n; o \leq Q_h\}.$$

where Q_n (Q_h) defines the occupancy from which the states in the redial (retrial) orbit are aggregated. In this case states of the form $(\cdot, Q_n, \cdot)$ represent the situation where at least Q_n users are in the redial orbit. Likewise the states of the form $(\cdot, \cdot, Q_h)$ represent the situation where there are Q_h or more users in the retrial orbit. Due to that aggregation two new parameters for each orbit are introduced. The parameter M_n denotes the mean number of users in the redial orbit conditioned to those states where there are at least Q_n users in the orbit, i.e., $M_n = E(m|m \geq Q_n)$. The probability that after a successful redial the number of users in the redial orbit does not drop below Q_n is represented by p_n. For the retrial orbit the parameters M_h and p_h are defined analogously.

The global balance equations, the normalization equation and equations for parameters M_n, p_n, M_h, p_h form a system of simultaneous non-linear equations, which can be solved using, for instance, the iterative procedure shown in [14].

3.3 Truncation and Generalization (TNR)

While the two previous approximations consider a finite truncated method for each retrial orbit, this method considers the use of a generalized truncated method in one of the two orbits. Obviously, we cannot use a generalized method for both orbits as the resulting model would not be solvable. For this reason, we have applied a generalized truncated method for the automatic retrial orbit and a Truncation (T) for the redial orbit. The method chosen for the retrial orbit is the method proposed by Neuts and Rao, denoted as NR, in [15]. This method is based on the homogenization of the model beyond a given level Q_h, which supposes to restrict the maximum automatic retrial rate, i.e.,

$$\mu_{ret}(o) = \begin{cases} o\mu_{ret} & \text{if } o < Q_h \\ Q_h\mu_{ret} & \text{if } o \geq Q_h \end{cases}$$

Therefore, the resulting state space is defined by

$$\mathcal{S} := \{(k, m, o) : k \leq C; m \leq Q_n; o \in \mathbb{Z}_+\}$$

With these two approximations we have to solve a system whose state space presents two finite dimensions and an infinite one, being the infinite dimension homogeneous beyond a given level Q_h. So, we can solve the resulting system and obtain the steady-state probabilities making use of the matrix-geometric solutions for stochastic models proposed by Neuts in [8].

4 Value Extrapolation

All the approximate methods described in the previous sections compute the steady-state probabilities using the balance equations. Very recently, however, an alternative approach for evaluating infinite state space Markov processes has been introduced by Leino et al. [5]. This approach, named Value Extrapolation (VE), does not rely on the probability of being in a certain state, but on a new metric called relative state values, that appear when we consider the system in its MDP setting. Formally, an MDP can be defined as a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P}$ is a state transition function and $\mathcal{R}$ is a revenue function. The state of the system can be controlled by choosing actions a from $\mathcal{A}$, influencing in this way the state transitions. The transition function $\mathcal{P} : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}_+$ specifies the transition rate to other states when a certain action is taken at a given state. The first characteristic of VE is the necessity of the definition of a revenue function that must be a function of the system state, i.e., $r(s)$. Following the definition of the revenue function for every state, we will also have a mean revenue rate of the entire process (r), which will be the performance metric we want to compute.

Once the MDP framework as well as the revenue function are specified, we are able to define the relative state values. It is obvious that after performing an action in state s the system will collect a revenue for that action ($r(s)$), but, as the number of transitions increases, the average revenue collected converges to r. The relative state value ($v(s)$) indicates the difference between the total revenue incurred when the system starts at state s and the total revenue incurred in a system for which the cost rate at all states is r. If we denote by t_n the time instants in which there is a change in the system state, then

$$v(s) = E\left[\sum_{n=0}^{\infty} (r(S(t_n)) - r) \Big| S(t_0) = s\right].$$

The equations that relate revenues, relative state values, and transition rates are the Howard equations defined by:

$$r(s) - r + \sum_{s'} q_{ss'}(v(s') - v(s)) = 0 \quad \forall s.$$

There will be as many Howard equations as number of states, $|\mathcal{S}|$. The number of unknowns will be the $|\mathcal{S}|$ relative state values plus the expected revenue r, i.e, $|\mathcal{S}| + 1$ unknowns. As only the differences in the relative values appear in the Howard equations, we can set $v(\mathbf{0}) = 0$, so we will have a solvable linear system of equations with the same number of equations as unknowns.

However, a finite number of Howard equations are needed to solve the system and, therefore, we need to truncate the state space to $\hat{\mathcal{S}}$. Whereas the traditional truncation consists of doing $q_{ss'} = 0 \quad \forall s' \notin \hat{\mathcal{S}}$, VE performs a more efficient truncation. Basically, VE considers the relative state values outside $\hat{\mathcal{S}}$

that appear in the Howard equations as an extrapolation of some relative state values inside $\hat{\mathcal{S}}$. The objective of VE is to find a function $f(s)$ that interpolates some points $(s, v(s))$ for $s \in \hat{\mathcal{S}}$ so that it approximates also $(s, v(s))$ for $s \notin \hat{\mathcal{S}}$. It is important to choose a fitting function, $f(s)$, that makes the Howard equations remain a closed system of linear equations. The most common fitting functions that acomplish that fact are the polynomials. We can use all $(s, v(s))$-pairs of the state space into the fitting procedure —global fitting— or only a subset $(\mathcal{S}_f)$ of them —local fitting. The choice of $\mathcal{S}_f$ will highly depend on the relative state value we want to extrapolate. Note also that function $f(s)$ and set $\mathcal{S}_f$ need to be chosen so that parameters have unambiguous values, i.e., in the case of choosing a polynomial as the fitting function, the number of different $(s, v(s))$-pairs in $\mathcal{S}_f$ has to be equal or greater than the number of coefficients in the polynomial. Note that if the relative values outside $\hat{\mathcal{S}}$ were correctly extrapolated, the results obtained by solving the truncated model would be exact.

4.1 Howard Equations of the System

To obtain the Howard equations for a certain state of the system under study, we can classify these states into four different cases depending on the number of sessions being served (k). We next describe such cases and their corresponding Howard equations.

1. $\mathbf{k < L}$: states in which both new sessions and handovers are accepted. The transition rates that go out from these states are represented in Fig. 2. Therefore, the Howard equations related to these states are:

$$r(k,m,o) - r + \lambda[v(k+1,m,o) - v(k,m,o)] + k\mu[v(k-1,m,o) - v(k,m,o)] +$$
$$+ m\mu_{red}[v(k+1,m-1,o) - v(k,m,o)] + o\mu_{ret}[v(k+1,m,o-1) - v(k,m,o)] = 0.$$

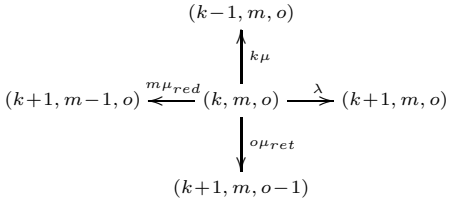


Fig. 2. Transition rates when $k < L$

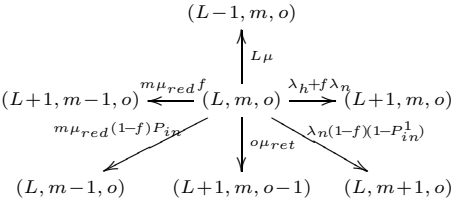


Fig. 3. Transition rates when $k = L$

2. $\mathbf{k = L}$: states in which handovers are accepted but new sessions are only accepted with probability $f = t - L$, where t is the parameter that characterizes the FGC admission control policy. Figure 3 represents the transition rates going out from these states, obtaining the next Howard equations:

$$r(L,m,o) - r + (\lambda_h + \lambda_n f)[v(L+1,m,o) - v(L,m,o)] +$$
$$+L\mu[v(L-1,m,o) - v(L,m,o)] + m\mu_{red} f[v(L+1,m-1,o) - v(L,m,o)] +$$
$$+m\mu_{red}(1-f)P_{in}[v(L,m-1,o) - v(L,m,o)] +$$
$$+o\mu_{ret}[v(L+1,m,o-1) - v(L,m,o)] +$$
$$+\lambda_n(1-f)(1-P_{in}^1)[v(L,m+1,o) - v(L,m,o)] = 0.$$

3. $\mathbf{L < k < C}$: states where handovers are accepted but new sessions are blocked, as shown in Fig. 4. That leads to the Howard equations:

$$r(k,m,o) - r + \lambda_h[v(k+1,m,o) - v(k,m,o)] + k\mu[v(k-1,m,o) - v(k,m,o)] +$$
$$+m\mu_{red}P_{in}[v(k,m-1,o) - v(k,m,o)] + o\mu_{ret}[v(k+1,m,o-1) - v(k,m,o)] +$$
$$+\lambda_n(1-P_{in}^1)[v(k,m+1,o) - v(k,m,o)] = 0.$$

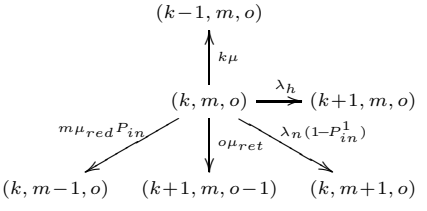


Fig. 4. Transition rates when $L < k < C$

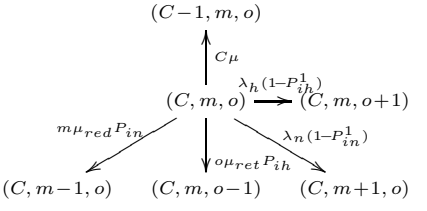


Fig. 5. Transition rates when $k = C$

4. $\mathbf{k = C}$: states where both new sessions and handovers are blocked, being the transition rates as shown in Fig. 5 and their corresponding Howard equations:

$$r(C,m,o) - r + \lambda_h(1-P_{ih}^1)[v(C,m,o+1) - v(C,m,o)] +$$
$$+C\mu[v(C-1,m,o) - v(C,m,o)] + m\mu_{red}P_{in}[v(C,m-1,o) - v(C,m,o)] +$$
$$+o\mu_{ret}P_{ih}[v(C,m,o-1) - v(C,m,o)] +$$
$$+\lambda_n(1-P_{in}^1)[v(C,m+1,o) - v(C,m,o)] = 0.$$

4.2 Revenue Function

As performance parameters are not computed from the steady-state probabilities as usual, it is important to explain more carefully how they are computed. For that purpose we must set the inputs $r(s)$ in the Howard equations properly in order to ensure that the revenue rate of the entire process r is equal to the performance parameter we want to compute. In a nutshell, r will be the parameter we want to compute if we let $r(s)$ to be the value of that parameter when the system is in state s. Table 2 gives several examples on how $r(s)$ can be set in order to obtain the performance parameters under study.

Table 2. Revenue function definition

Parameter	Value
P_b^h	$r(k,m,o) = 1$ for $k = C$, $\forall m$, $\forall o$
	$r(k,m,o) = 0$ otherwise
P_b^n	$r(k,m,o) = 1 - f$ for $k = L$, $\forall m$, $\forall o$
	$r(k,m,o) = 1$ for $k \geq L$, $\forall m$, $\forall o$
	$r(k,m) = 0$ otherwise
N_{ret}	$r(k,m,o) = o$ $\forall k$, $\forall m$, $\forall o$
N_{red}	$r(k,m,o) = m$ $\forall k$, $\forall m$, $\forall o$

4.3 Polynomial Fitting and Solution

Note that in the system under study the number of states is infinite because both m and o can take any value in $\mathbb{Z}_+$, thus some truncation is needed. We have made a truncation similar to DT and DFM, obtaining a truncated state space defined by:

$$\hat{\mathcal{S}} := \{s = (k,m,o) : k \leq C; m \leq Q_n; o \leq Q_h\}.$$

Therefore, in the system under study, we have truncated the state space beyond a value of Q_n (Q_h) for the occupancy of the redial (automatic retrial) orbit. However, in the Howard equations of the truncated state space, relative state values of some states appear that do not belong to the truncated state space, being $v(C,m,Q_h+1)$ $\forall m$ and $v(k,Q_n+1,o)$ for $k \geq L$ and $\forall o$. Therefore, we must extrapolate these two sets of states to obtain a closed system of equations. We have used a $(n-1)$-th degree polynomial that interpolates the n points in $\{(j,v_j)|v_j = v(C,m,j), \forall m, Q_h - n < j \leq Q_h\}$ to extrapolate $v(C,m,Q_h+1)$. To extrapolate $v(k,Q_n+1,o)$ for $k \geq L$ we interpolate the p points in $\{(i,v_i)|v_i = v(k,i,o), k \geq L, Q_n - p < i \leq Q_n, \forall o\}$. Note that including value extrapolation neither increase the computational cost nor the number of Howard equations, remaining in $|\hat{\mathcal{S}}| = (C+1) \times (Q_n+1) \times (Q_h+1)$.

After some algebra, and using the Lagrange basis to reduce the complexity of the procedure, we obtain a simple closed-form expression for the extrapolated value of both sets

$$v(C,m,Q_h+1)^{(n)} = \sum_{j=0}^{n-1} (-1)^j \binom{n}{j+1} v(C,m,Q_h-j), \forall m,$$

and

$$v(k,Q_n+1,o)^{(g)} = \sum_{i=0}^{g-1} (-1)^i \binom{g}{i+1} v(k,Q_n-i,o), k \geq L, \forall o.$$

5 Results and Discussion

In this section a number of numerical examples are presented with the purpose of illustrating the capabilities and versatility of our model and the analysis

methodology. The numerical analysis is also aimed at assessing a comparison between the proposed methodology and previous approaches not only in terms of accuracy but also in terms of computation cost.

For the numerical experiments a basic configuration of the system is used and then the different parameters are varied. Thus, unless otherwise indicated, the value of the parameters will be those of the basic configuration: $C = 10$, $t = 9$, $\mu = 1$, $P_{ih}^1 = P_{in}^1 = 0$, $P_{ih} = P_{in} = 0.2$, and $\mu_{red} = \mu_{ret} = 1$. The values of λ_n and λ_h have been modified by means of the system load $\rho = \lambda/C\mu$, being $\lambda = \lambda_n + \lambda_h$ and taking $\lambda_h = 2\lambda_n$ in all cases. It must be noted that, due to the introduction of the impatience phenomenon modeled by P_{in}^1, P_{in}, P_{ih}^1, and P_{ih}, we will be able to consider values of $\rho > 1$.

5.1 VE Performance

The objective of this section is to study the performance of different extrapolation polynomials in a wide range of scenarios. Obviously, as stated in Section 3, for the system under study we are not able to compute the exact values of the most common performance parameters. For this reason, the first step is to assume that the exact value can be obtained choosing increasing and sufficiently high values of the truncation level. More specifically, we ran all methods presented in Section 3 and VE until the value of all the performance parameters under study had stabilized up to the 8th decimal digit.

In the system under study, there are two different truncation levels that must be specified, namely Q_n and Q_h. The purpose of this study will be to determine the pair (Q_n, Q_h) that makes the cardinality of the problem $((C + 1) \times (Q_n + 1) \times (Q_h + 1))$ as small as possible while a certain accuracy criterion is met. To fulfil these requirements we must define a direction of search to determine the desired (Q_n, Q_h) pair.

To avoid an exhaustive search to determine (Q_n, Q_h) we have used an algorithm similar to the one proposed in [16]. Our algorithm increase (Q_n, Q_h) along the diagonal until we obtain a system that fulfils the desired accuracy and later we decrease both parameters separately following descendent directions of the coordinate axis and finally take the best solution in terms of the cardinality of the problem. The rationale behind this last movement for only one of the two parameters (Q_n or Q_h) is the fact that, generally, $Q_n \neq Q_h$, and this cannot be accomplished only with the diagonal movement, so the solution with this last movement improves the initial diagonal movement.

In Table 3 we show the minimum complexity of the problem needed to fulfil a relative error lower than 10^{-4} for parameters P_b^n and P_b^h, for different loads (ρ) and reattempt rates ($\{\mu_{red}, \mu_{ret}\}$) and for different orders of the extrapolation polynomial.

Note that VEx denotes the use of an extrapolation polynomial of order x. Note also that the numbers shown in each cell represent the product $(Q_n+1)\times(Q_h+1)$ which defines the complexity and it is denoted by Ω, although the cardinality of the problem should also include the factor $(C+1)$. However, we have omitted this factor as it is common to all cases. Therefore, the best order for the extrapolation

Table 3. Minimum Ω to obtain relative errors lower than 10^{-4} in P_b^n/P_b^h

μ_{red}, μ_{ret}	ρ	VE1	VE2	VE3	VE4	VE5	VE6
{1,1}	0.4	25/30	**12/12**	16/16	25/25	36/36	49/49
	0.8	144/144	49/72	64/72	49/**35**	**36**/36	49/49
	1.2	484/506	342/342	240/**36**	**98**/120	121/132	99/120
{2,0.5}	0.4	20/25	**12/12**	16/16	25/25	36/36	49/49
	0.8	130/90	45/55	56/64	**36/30**	**36**/36	49/49
	1.2	-/-	432/336	280/170	**99/136**	126/144	135/168
{0.5,2}	0.4	20/25	**12/12**	16/16	25/25	36/36	49/49
	0.8	160/160	66/110	80/100	56/49	**36/42**	49/49
	1.2	-/-	-/-	400/-	154/189	**144/187**	162/198
{0.5,0.5}	0.4	25/30	**9/9**	16/16	25/25	36/36	49/49
	0.8	224/160	100/121	90/100	48/**35**	**36**/36	49/49
	1.2	-/-	-/-	-/-	**168**/280	195/**196**	441/378

polynomial will be the one that has the lowest Ω, which is in bold in the table. Moreover, we denote by "-" those cases in which the computer could not obtain a result because of lack of memory[1].

From the results in Table 3 we can conclude that there is not a clear choice in the order of the extrapolation polynomial that is able to get the lowest Ω in all cases. Neither the lowest nor the highest orders offer the best results. When the load is not high ($\rho = 0.4$), VE2 offers the lowest complexities, due to the fact that VE3-VE6 offer the result of the minimum Ω they require to work, e.g., to extrapolate with VE4 at least $Q_n = Q_h = 4$ is needed and therefore, the minimum Ω required to use VE4 is $(4+1) \times (4+1) = 25$. When the retrial orbits are more heavily loaded, VE4 is a good choice, as it offers low values of Ω. Therefore, hereafter we will use the polynomial of order 4 (VE4) and we will simply denote it as VE.

5.2 Comparison among Different Methods

Accuracy: The objective of this section is to compare the performance of VE with DT, DFM, and TNR. In Table 4 we show the minimum values of Ω needed to obtain a relative error lower than 10^{-4} for N_{red}. The results for the rest of performance parameters have been omitted as N_{red} is usually the worst case for all methods and results are found to be qualitatively equivalent for all performance parameters. We show in bold the best results, i.e., those that offer the minimum complexity Ω. Results show that VE clearly outperforms classical methods as it needs a much lower value of Ω to achieve the desired accuracy in all the scenarios under study. Moreover, and what is probably more important, there are some scenarios where VE is the only method that is able to get a result due to the complexity of those scenarios produced by having low reattempt rates.

[1] Results have been obtained using Matlab running in an Intel Core 2 Quad Q6600 with 4GB RAM memory.

Table 4. Minimum Ω to obtain relative errors lower than 10^{-4} in N_{red}

	$\mu_{red}, \mu_{ret} = \{1, 1\}$					$\mu_{red}, \mu_{ret} = \{2, 0.5\}$					$\mu_{red}, \mu_{ret} = \{0.5, 5\}$					$\mu_{red}, \mu_{ret} = \{0.5, 0.5\}$				
ρ	0.4	0.6	0.8	1.0	1.2	0.4	0.6	0.8	1.0	1.2	0.4	0.6	0.8	1.0	1.2	0.4	0.6	0.8	1.0	1.2
DT	64	143	324	550	930	56	132	304	522	-	54	120	264	-	-	63	180	528	-	-
DFM	48	72	208	360	324	49	100	176	378	-	45	98	198	-	-	56	126	352	-	-
TNR	48	91	180	400	651	48	99	182	196	640	36	90	192	-	-	54	135	240	-	-
VE	**25**	**25**	**35**	**110**	**196**	**25**	**25**	**35**	**108**	**204**	**25**	**25**	**60**	**66**	**161**	**25**	**25**	**45**	**195**	**396**

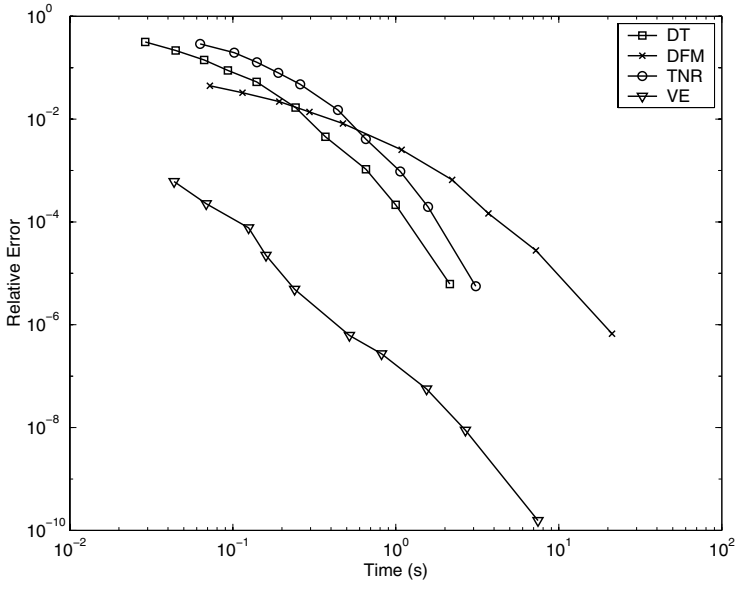


Fig. 6. Computation time for different methods

Computation cost: Although it is shown that VE clearly outperforms the other methods in terms of accuracy, it is also interesting to study their associated computation cost. From a practical perspective, it is more interesting to consider accuracy along with computation time. Figure 6 shows a joint representation of both parameters. As the figure shows, VE yields much higher accuracy than any other method for a given computation time. Results should be interpreted carefully, because computation cost highly depend on the algorithm used to solve the resulting system of equations. More concretely, in order to compute matrix $\mathbf{R}$ that appear in TNR we have used the logarithmic reduction algorithm as proposed in [17, Section 8.4], using a precision of 10^{-6} for the iterative procedure. Moreover, for solving the systems obtained with the DT, DFM, and TNR methods we have made use of the efficient algorithm described in [11] that takes advantage of the block-tridiagonal structure that presents the infinitesimal generator. Unfortunately, the linear system of equations obtained in VE has no

longer such a block-tridiagonal structure, and therefore, we must use a more general algorithm. More concretely, we have used LU factorization.

It can be seen that in the system under study the computation times needed for any of the methods are not very high from a human point of view. For that reason, the time results should be compared qualitatively, as the time units may be different from just seconds when we solve more complex systems or when we have to solve the basic retrial system several times —for example to balance the incoming handover rate to the outgoing handover rate, as shown in [18]—.

6 Conclusions

In mobile communication systems like cellular networks, Mobile IP or the recently defined IEEE 802.16e and IEEE 802.20 networks, mobile operators must guarantee seamless mobility to its customers. In these networks, repeated attempts occur due to user redials when their session establishments are blocked and also due to automatic retries when a handover fails. The Markovian model describing such a complex network is a multiserver retrial system that presents space-heterogeneity along two infinite dimensions. To the best of our knowledge, all the methods studied in the literature to solve these systems are based on their steady-state probabilities. In this paper, we propose an alternative method based on a different metric: the relative state values and the Howard equations that relate them.

We have compared the proposed method with the most well-known approaches appeared in the literature so far. The results show that the proposed method greatly outperforms previous approaches not only in terms of accuracy, but also in terms of computation cost. Moreover, we have shown that in some scenarios the proposed method is the only one that is able to guarantee a certain accuracy. For all those reasons the proposed method is highly recommendable to solve this type of systems.

Acknowledgements

This work was supported by the Spanish Government (30% PGE) and the European Commission (70% FEDER) through projects TSI2007-66869-C02-02 and TIN2008-06739-C04-02.

References

1. Mouly, M., Pautet, M.B.: The GSM system for mobile communications. Published by the authors (1992)
2. Onur, E., Deliç, H., Ersoy, C., Çaglayan, M.U.: Measurement-based replanning of cell capacities in GSM networks. Computer Networks 39, 749–767 (2002)
3. Artalejo, J.R., Pozo, M.: Numerical calculation of the stationary distribution of the main multiserver retrial queue. Annals of Operations Research 116(1–4), 41–56 (2002)

4. Domenech-Benlloch, M.J., Gimenez-Guzman, J.M., Pla, V., Martinez-Bauset, J., Casares-Giner, V.: Generalized Truncated Methods for an Efficient Solution of Retrial Systems. Mathematical Problems in Engineering, Article ID 183089 (2008)
5. Leino, J., Penttinen, A., Virtamo, J.: Flow level performance analysis of wireless data networks: A case study. In: Proceedings of IEEE ICC, vol. 3, pp. 961–966 (2006)
6. Khan, F., Zeghlache, D.: Effect of Cell Residence Time Distribution on the Performance of Cellular Mobile Networks. In: Proceedings of IEEE VTC 1997, pp. 949–953 (1997)
7. Ramjee, R., Nagarajan, R., Towsley, D.: On optimal call admission control in cellular networks. Wireless Networks Journal (WINET) 3(1), 29–41 (1997)
8. Neuts, M.: Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach. The Johns Hopkins University Press (1981)
9. Falin, G., Templeton, J.: Retrial Queues. Chapman & Hall, Boca Raton (1997)
10. Wilkinson, R.I.: Theories for toll traffic engineering in the USA. The Bell System Technical Journal 35(2), 421–514 (1956)
11. Servi, L.D.: Algorithmic solutions to two-dimensional birth-death processes with application to capacity planning. Telecommunication Systems 21(2–4), 205–212 (2002)
12. Domenech-Benlloch, M.J., Gimenez-Guzman, J.M., Martinez-Bauset, J., Casares-Giner, V.: Efficient and accurate methodology for solving multiserver retrial systems. IEE Electronic Letters 41(17), 967–969 (2005)
13. Marsan, M.A., Carolis, G.D., Leonardi, E., Cigno, R.L., Meo, M.: Efficient estimation of call blocking probabilities in cellular mobile telephony networks with customer retrials. IEEE Journal on Selected Areas in Communications 19(2), 332–346 (2001)
14. Gimenez-Guzman, J.M., Domenech-Benlloch, M.J., Pla, V., Casares-Giner, V., Martinez-Bauset, J.: Guaranteeing Seamless Mobility with User Redials and Automatic Handover Retrials. Journal of Universal Computer Science 14(10), 1597–1624 (2008)
15. Neuts, M., Rao, B.: Numerical investigation of a multiserver retrial model. Queueing systems 7, 169–190 (1990)
16. Artalejo, J.R., Pla, V.: On the impact of customer balking, impatience and retrials in telecommunication systems. Computers & Mathematics with Applications 57(2), 217–229 (2009)
17. Latouche, G., Ramaswami, V.: Introduction to Matrix Analytic Methods in Stochastic Modeling. In: ASA-SIAM 1999 (1999)
18. Marsan, M.A., Carolis, G.D., Leonardi, E., Cigno, R.L., Meo, M.: How many cells should be considered to accurately predict the performance of cellular networks? In: Proceedings European Wireless (1999)

Study of the Path Average Lifetime in Ad Hoc Networks Using Stochastic Activity Networks

Teresa Albero-Albero[1], Víctor-M. Sempere-Payá[1], and Jorge Mataix-Oltra[2]

[1] Instituto Universitario Mixto Tecnológico de Informática,
Universidad Politécnica de Valencia,
Camino de Vera s/n, 46022 Valencia, España
{maalal0,vsempere}@dcom.upv.es
[2] Universidad Politécnica de Cataluña (UPC)
Av. del Canal Olímpic, s/n, 08860 Castelldefels, España
jordi.mataix@upc.edu

Abstract. The supervision of industrial processes requires the exchange of information in real time between users and control systems. Users may be moving around a working area and need to consult information to supervise a particular process. Therefore, it is important to study the characteristics and stability of the paths to determine which services can be offered. In this paper, the effect of mobility on duration and stability of the links in an ad hoc network is analysed using stochastic activity networks. The ad hoc network is made up of six mobile nodes where the routing protocol is AODV. This study shows the path average lifetime which enables the evaluation of which type of services can be offered by the network.

Keywords: SAN (stochastic activity networks), modelling, MANET, routing protocols, route maintenance, path average lifetime.

1 Introduction

In industrial environments the need to exchange information between mobile users within a working area is becoming increasingly common. The services offered to a user include:

- Information on process alarms.
- Access to previously stored control images [1], [2] or images related to a previous event.
- General images of the plant and images of specific processes which need to be monitored or controlled in real time by cameras installed in the plant.

In this paper we have studied the link performance which will in turn enable us to understand which services can be supported by an ad hoc network with sufficient quality and under which conditions.

A path is created when two users are in communication. A path can include two or more links. The length of the path is the number of links. Therefore,

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 71–88, 2009.

link stability is crucial when generating a path between users. The protocol performance depends on the duration of a path between the source and the destination (path average lifetime).

In this paper we analyze the effect of the number of hops, the transmission range and the speed of the mobility on the path average lifetime. Node mobility is the major factor affecting the performance of the routing protocols. Since a link break from a node movement invalidates all the routes containing this link, alternate routes have to be discovered once the link is detected as broken. Because of this, we have studied how node mobility affects these paths, causing breaks in the links. The path average lifetime enables us to identify which services can be offered on the ad hoc network designed, always taking into account that the new route discovery will create a flood of routing requests and extended delay for packet delivery.

This paper presents a study, in which formal models were used to analyse the effect of mobility in an ad hoc network with six mobile nodes on the duration and stability of the paths in order to determine how the services were affected. The scenario to which these results may be applied does not require a large number of nodes. Six nodes are sufficient to cover the working area, and there are not usually more users needing to exchange information in these circumstances.

Although these results were obtained from a simulation, our aim is to create a scenario as close to reality as possible. Our study is based on the routing protocol AODV (Ad Hoc On-Demand Distance Vector Routing) [3], which is one of the routing protocols under active development inside the IETF MANET working group [4]. AODV is together with OLSR (Optimized Link State Routing) [5] the most mature routing protocol from the implementation standpoint. It is for this reason that they are the two most studied protocols. In [6], [7] are described real experiments where the performance of these two routing protocols is compared, using a number of nodes ranging from 5 up to 12 nodes (laptops and PDAs). Other experimental evaluations have been carried out with a similar number of nodes, in [8] 5 laptop computers were used to study AODV routing protocols and OLSR protocols. Previous studies [9] have shown the existence of an Ad Hoc horizon (2-3 hops and 10-20 nodes) after which the benefit of multi-hop ad hoc networking disappears. As Conti states in [10], it is unrealistic to centre the research on networks with hundreds of mobile nodes involved in CBR (constant bit rate) data transfers.

AODV is a reactive protocol that minimizes the number of route broadcasts by creating routes on-demand. Route discovery is initiated on-demand, the route request (RREQ) is forwarded by the source node to the neighbours, and so on, until either the destination or an intermediate node with a fresh route to the destination, is located. The response to the route found is sent via a RREP packet (route replay). This route can be single hop, which is a direct communication or multi-hop when neighbouring nodes are necessary to reach the destination.

In section 2 previous results are presented and in 3 the scenario and the various parts of the model are explained. The measurements and results are presented in section 4, while the conclusions and future work are shown in section 5.

2 Previous Results

In previous works by the authors, stochastic activity networks have been used to create formal models that enable the study of mobility and reachability between nodes [11]. In these models the tool used was UltraSAN [12] and now the tool used is Mbius 2.1 [13], [14] both of which supports the specification of SAN [15] models.

With our previous models the probability of the source reaching the destination in an Ad Hoc network was studied in function of radio transmission range [11], determining direct and indirect communications and failed attempts at communication in which the destination was unreachable. In multi-hop[1] communications, with a radio range of more than 150m successful communications began to decrease; direct communications with distances greater than 150m will probably[2] be successful. In other words, 150m was identified as the range for which the number of multi-hop communications reached maximum[3] value. The radio range that offers this maximum value of multi-hop communications was previously difficult to identify and with the SAN tool it has been obtained in a simple manner.

Later, a more detailed study was carried out on multi-hop communications [16], [17] dividing these according to the number of hops in each path established. We observed that communications with two or three hops were the most numerous, confirming the findings of Tschudin et al. in their studies [9]. Furthermore most MANET routing protocols focus on minimizing the hop count of the chosen path [18]. The number of hops corresponds with the number of times a packet must be transmitted and received to reach its destination. Each additional transmission has some consequences; a longer path consumes additional bandwidth and additional hops add more delay due to the additional buffering, contention, and transmission time required. For this reason, in the models preference is given to shorter routes, thereby minimizing the number of links that may break causing a path failure. From the percentages of multi-hop communications and according to the number of nodes participating in the path, we were able to make an estimation of the potential energy savings obtained by using this type of communication.

[1] The terms direct or single-hop communications and indirect or multi-hop communications are used interchangeably.

[2] In the models designed, preference was given to communications with the lower number of nodes (just as an AODV behaves), and therefore direct communications have preference over multi- hop, and it is because of this that with a certain value of radio range an inflection point occurs after which there is a decrease in multi-hop communication giving way to a greater percentage of direct communications between source and destination.

[3] The most significant characteristic of ad hoc networks is the use of neighbouring nodes to reach the destination, and this is denominated multi-hop communication. It is always possible to choose a range of radio ranges which enables us to obtain a balance between energy savings through use of multi-hop communication against single hop and a satisfactory number of communications obtained.

In light of these results, we consider it interesting to study more deeply the performance ad hoc networks. SAN enables us to modify and widen these models, in a simple way and without having to redesign them. New submodels that represent the working of the AODV protocol at the moment when the requested route is obtained and when a lost route is recovered can be added.

3 Models

In this section, firstly we describe the scenario in which the ad hoc network was incorporated, along with the parameters used in its design. Secondly there is a description of each of the subnetworks that form the formal model of the ad hoc network in question, in which the AODV was the routing protocol used. The objective of these formal models is the evaluation of the effect of mobility on the duration and stability of the routes in an ad hoc network.

3.1 Scenario

An area of 350x350m^2 is large enough to cover most industrial installations. The shape of the area is determined by the use of hexagonal cells. This type of cell facilitates the representation of movement of nodes and their coverage radio range. Six nodes were distributed (A, B, C, D, E, F) in the area, see Fig. 1. In order to be sure that the initial position of the nodes did not affect the results, different tests were carried out. This initial position was varied, without variations in the results.

Regarding the number of nodes used in models, the scenario to which these results may be applied does not require a large number of nodes, because this number is sufficient to cover the working area. For this reason, in our case we considered 6 to be the maximum number of mobile nodes. Moreover it is known that a high density can cause traffic problems and reduce the efficiency of the channel usage [10]. Furthermore, experiments in real scenarios are made up of a few nodes, see [19], in which a multi-hop wireless ad hoc network was constructed in a testbed of 8 nodes, 2 of them fixed, over an area of 700x300m^2 and [7] where experiments in string topology were set up in an open field about 300 meters long. In [6], [7] real experiments were carried out, using a number of nodes ranging from 5 up to 12 nodes.

As the area used is divided into hexagonal cells of the same size, see [20], the probability of one node to visiting its neighbouring cells has an assumed value, p = 1/6. The mobility model used in the results presented in this paper is a random walk model.

We assume the time that a mobile node stays in a cell and the number of communication attempts. These have an exponential time with mean value $1/\lambda$m and $1/\lambda$c respectively [20]. The values of search rate and movement rate have been chosen to adjust with real values of movement of the nodes [21], [22]. To this end, considering that every cell is equivalent to 50m, we can calculate the mean velocity of the users and this enables choose the most appropriate movement rate.

In the experiments, session times of 20s, 1 and 3 minutes were used. The authors have chose typical session times used in the supervision of industrial processes. In a real scenario, during this time the user could see control or multimedia information. Depending on the service requested, it is logical that the user will have to wait for differing periods of time. For example, if the user requests detailed information on a process from a sensor, 20s can be considered long enough to obtain and evaluate this information. However, when the user wants to see control images from the installation or of a process, more time will be needed to see and evaluate the images.

Another parameter used in these models is the radio transmission range, (R) with values between 100m and 200m. In the experiments this was a nominal range, with no variation. Two nodes can establish a connection when the distance between them is equal to or less than R. In our case for example, as each hexagonal cell measures 50m across, one node with a radio range of 100m can connect with all those nodes situated within the 2 rings around the cell in which the node is located. In Fig. 1 the lined cells represent the radio coverage of 100m from the node located in cell (1, 1).

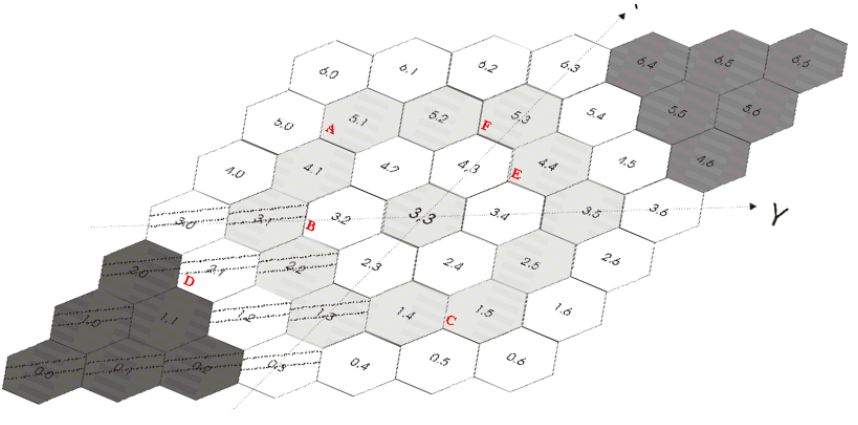


Fig. 1. Working area, 350x350m^2, numbering of cells and start position of mobile nodes

It is known that the characterization of the wireless channel is one of the critical points in MANET simulation modelling. Although ideally we would like the scenario to be as realistic as possible, certain assumptions were necessary. For example, no link layer effects, such as HELLO packet losses were considered, although it has been shown that this has a real effect on the link establishment in MANETs [23]. It can also be assumed that interference from neighbouring nodes will be nil. The hidden[4] node problem has not been considered. The exposed[5]

[4] Those nodes that cannot establish a connection directly between each other could still be transmitting messages simultaneously to a common neighbour on the same frequency.

[5] A node near an active sender is ineligible to send or receive.

node has also not been considered. We assume that links are symmetrical and this is far from reality. Finally, it is important to note that there is no traffic on the network, and so there are always resources available. We know that one drawback with models is that when simplifications and assumptions are introduced; they sometimes mask important characteristics of the real protocols performance. For this reason, so that the performance of the models used is as realistic as possible, we are working on the introduction of transmission errors in the model along with heavy loads on the ad hoc networks being studied.

3.2 Characteristics of the Models

Most MANET studies are base on simulation tools. The most popular simulators used in Ad Hoc networks are OPNET, ns-2 and Glomosim, but these are not the only valid tools for studying this type of network. In this paper, we use the power of stochastic activity networks (SANs) to observe the performance of ad hoc networks based on submodels already designed in previous studies.

Stochastic activity networks are a stochastic extension of Petri nets to define temporary characteristics with statistical parameters. Colored Petri nets [24] and Fuzzy Petri nets [25] have been used to study mobility in ad hoc networks. UltraSAN is used by different authors [20] to model mobility in mobile terminals. Mbius, the successor of UltraSAN, is used for the creation of the formal models whose results are presented in this paper.

The models designed are formed by five submodels:

- The "search" submodel shows the attempt of communication between two nodes.
- The "position" submodels represent the position of every node in the area and its movement through it. There is one position submodel for every node in the network.
- The "recover route" submodel studies whether a path remains active after a movement. In this submodel if a path is lost, a new one is sought.
- The "time" submodel.
- And the "time to recover" both of which submodels are necessary to find the average lifetime of a path.

The five submodels are interconnected, sharing some of their elements. In Fig. 2, there is a brief outline plan of the global functioning, in which this interconnection can be seen.

The "search" submodel, marked as (1) in Fig. 2, is used when there is a communication attempt between source and destination, and the source node sends a RREQ packet. The model is designed in such a way that the source and destination nodes were always the same to simplify the programming. Knowing the position of every node[6] in the model, we find out if the communication with

[6] To find a route from a source node to a destination, the source node in the Petri net model should have its neighbours identifications to send broadcast messages. Note that it is not necessary for nodes in a real MANET to know their neighbours, [24].

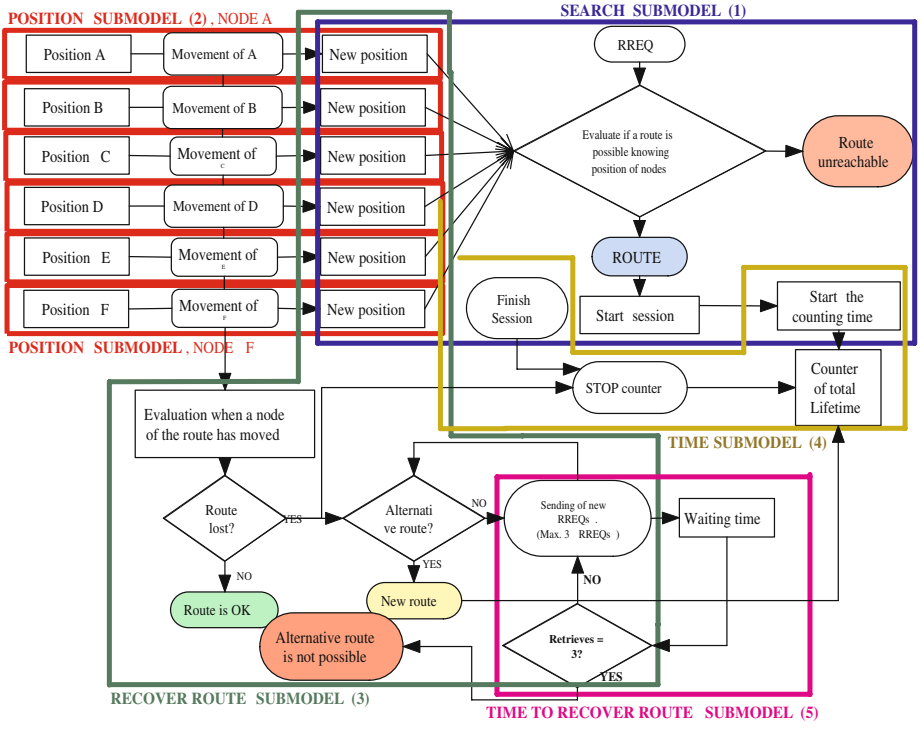


Fig. 2. Diagram of how the model works. Interrelation between submodels.

the destination is direct, indirect (we obtain the number of hops) or if it is not possible to communicate, using the supposition that there are no errors and no traffic. In this calculation, the exact[7] path is obtained. With no prior knowledge, one path is equally likely to be as good as another, so in the model, the first path found is chosen, always with the least number of nodes. At this moment, the session is initiated because the user has requested a communication with the destination and the request has been satisfied. The start of service means the start of the count of the time that the route is active; the counter is situated in the "time" submodel (4). This count will stop when the user terminates the session or when the connection is broken, see "recover route" submodel (3).

In the "search" submodel, see Fig. 3, if after the first RREQ there has not been response, the process established by the AODV is initiated whereby there are waiting times before the next RREQ packets are sent. Therefore, before a destination is given up as unreachable, the route is requested up to three times. If after the sending of the first RREQ, a positive response is not received, the

[7] Discovering the exact path and storing it is very difficult in terms of programming, but is necessary in order to be able to establish whether, after the movement of a node forming part of a path, the path has been affected.

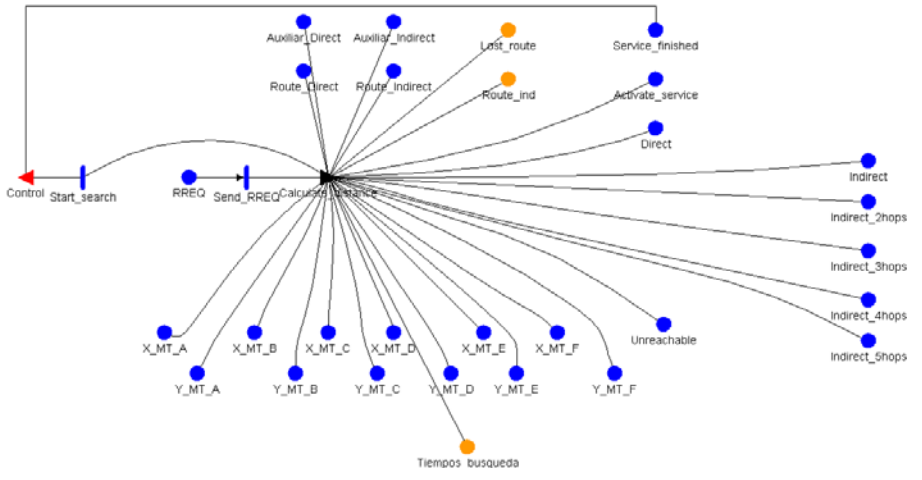


Fig. 3. Search submodel designed with Mbius

following RREQ packet is not sent until 0.4s has elapsed. Once this time has elapsed, if no response has been received, a new RREQ packet will be sent. In the same way, if there is still no response after this second sending, a new packet will be sent, this time after 0.8s. Finally, last request will be sent after 1.6s further waiting time. If no route is possible, the destination is given up as unreachable and the user cannot begin the session.

There is a "position" submodel, marked as (2) in Fig. 2, for every node in the network. This submodel evaluates the position of the node and its movement, obtaining the new position[8], which in turn depends on the angle of movement. If the node changes cell, the distance to the other nodes of the network is obtained. Also, if the node that has moved belongs to the active route, it is necessary to check if the movement has caused a break in the route or if the route remains active. In Fig. 4, the position submodel of node A designed with Mbius is presented.

The evaluation of the route after a node moves is done in the "recover route" submodel, marked as (3) in Fig. 2. Knowing the distance between nodes and the current route, it is checked whether the movement of a node belonging to the route has caused a break. There are two possibilities:

- A break has not occurred. The route remains the same as before although the node is not in the same position. This means that the node that has moved is still within range of its neighbours in the route.
- A break has occurred. The distance between the node that has moved and its neighbours is now greater than the radio transmission range; therefore new calculations are made to find an auxiliary route if possible. The auxiliary

[8] Note that at the start of the simulation (Fig. 1) every node is alone in this cell but after its movements two or more nodes can share cell.

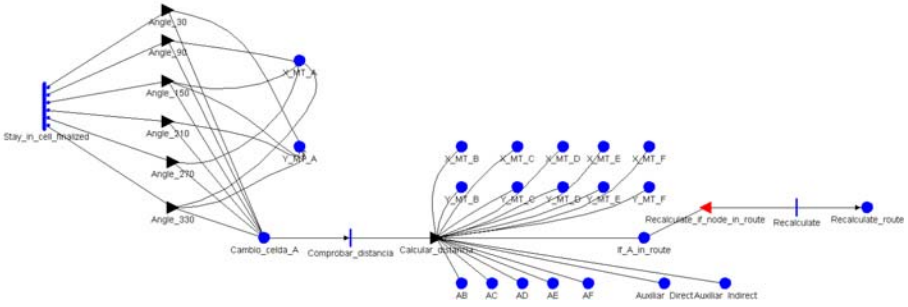


Fig. 4. Position submodel of node A designed with Mbius

route chosen will always be the shortest in the case that there is more than one possibility. In order to find an auxiliary route the route request (RREQ) is sent up to three times, repeating the same mechanisms as in the "search" submodel. This is done in the "time to recover route" submodel. The model takes into account the time needed to find the route and the number of nodes which form it.

In Fig. 5, the "recover route" submodel designed with Mbius is presented. We have highlighted the different blocks that make it up; therefore it can be more easily understood. It is important to state that "recover route" submodel programming is very complex (mainly the programming of some output gates), and for this reason we have included the flow chart to show how it works with the others submodels, Fig. 2. This complexity is due to the decision to create a "position" submodel in order to know the exact position of each node in the network. Without knowing the exact position in the network, it is not possible to know the exact route when a communication is requested between origin and destination, and consequently it is impossible to know when a movement will mean a loss of path. Authors such as Murata et al. [24], have previously studied, through simulations, how mobility affects the performance of the AODV, but without knowing the exact topology of the network in question, they conclude that it is not easy to build a CPN (colored Petri net) of a MANET because a node can move in and out of its transmission range and thus the MANETs topology dynamically changes. Therefore, they propose a topology approximation to address this problem of mobility. According to the authors, it is possible to model a MANET without information on its exact graph structure, but this makes it impossible for them to study a break in the path and its recovery. Other authors [25] have also analysed the AODV with a variant, but in their algorithm they dont need to compute the ad hoc network topology and they only need the information of neighbouring nodes for each node. However, they use the mechanism of Fuzzy Petri net to find a route with the highest reliability but they have not studied what happens after a route is obtained.

The "time" and "time to recover route" submodels, are marked such as (4) and (5) respectively in Fig. 2. These submodels together with the elements of the

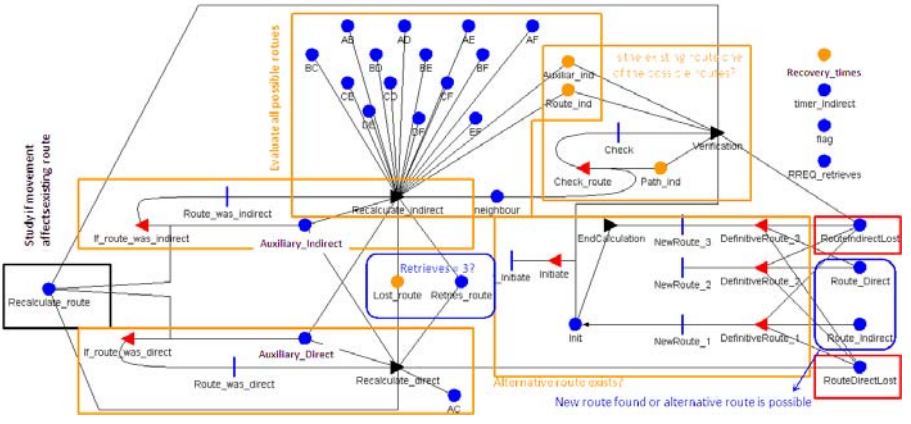


Fig. 5. Recover route submodel designed with Mbius

other submodels with which they interact are necessary to obtain the average time that a route remains active (average path lifetime). The "time to recover route" submodel (5) interacts with "recover route" (3), and the same as when a route is lost due to a movement of a node ("position" submodel), a recovery process is initiated. This process is similar to the one carried out when an initial route is required between source and destination, as there are also three attempts to find an alternate route. In Fig. 6 "time to recover route" submodel is presented.

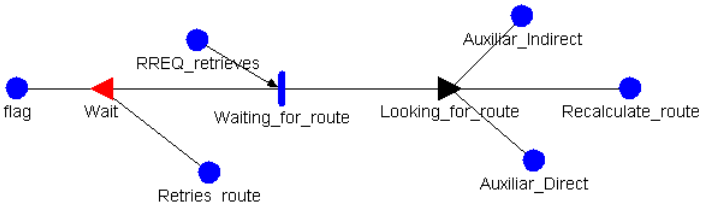


Fig. 6. Time to recover route submodel designed with Mbius

The submodel "time" (4) in Fig. 2, measures the time during which the route is active, whether this is in direct or indirect form. If a path is found in the "search" submodel (1), the session is initiated and in the "time" submodel the counting time is initiated too. In reality, there will finally be the sum of all the times when communication has been possible (total lifetime) in the counter, and knowing the number of routes that have been obtained (direct or indirect) we can obtain the average time that the routes have been active (average link lifetime).

Because of this, in this submodel it is necessary to know:

- When the direct or indirect route has been possible; this initiates the session and the total lifetime counter.
- When the route has been lost (recover route submodel), this stops the counter.
- And when the user finishes the session, at this moment the counter is stopped. The "time" submodel designed is shown in Fig. 7.

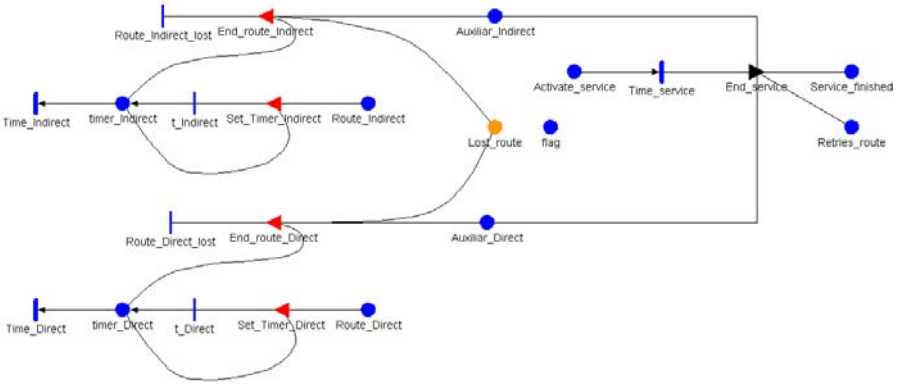


Fig. 7. Time submodel designed with Mbius

4 Measurements and Results

In this section the results using the models designed with 6 nodes according to the design described in section 3 are shown.

- Firstly, the lost and recovered paths are evaluated. Lost and recovered paths are also divided into direct and indirect.
- Secondly, average path lifetime is evaluated. This is the average time that a path remains active.

The programming of some elements of the model is very complex. For this reason, they were resolved through simulation rather than analytically.

Reward formalisms are functions that measure information about the system being modelled. Currently, Mbius provides one reward formalism, performance variables. The reward variables used to measure the results are impulse rewards that can be used to count the number of times an action is executed during an interval of time. We have used 6000 time units (seconds), as simulation time. Rate reward variables are used to evaluate the number of tokens that have accumulated in certain places. Each reward variable was evaluated for a confidence level of 0.80 and a confidence interval of 0.1, that is, the average value of the result will not be satisfied until the confidence interval is within 10% of the mean

estimate 80% of the time. Each experiment was repeated 5 times to check the validity of the results.

The mobility rate (λm) has been chosen as 10/6, that is, 10 movements every 6 minutes. With this rate and knowing that the size of a cell in the area is 50m, the speed of movement of the nodes is: (10 movements x 50m / 6 minutes) = 1.38m/s. This is the speed used in all the experiments except for the representation of link average lifetime where other speeds have been used, [1.38, 5, 6, 7, 8, 9, 10, 15, 20] m/s. The values of λm used in the experiments to correspond with these speeds were: 0.1, 0.12, 0.14, 0.16, 0.18, 0.2, 0.3, and 0.4. This enables us to observe the evolution of the path average lifetime with the speed of the nodes and compare the results with those shown in [18]. Maximum speeds are chosen in order to test the routing protocol because maximum speeds result in frequent routing changes and test the abilities of the protocol to react. In fact, with the above-mentioned speeds we are not really considering a realistic scenario, but rather an extreme scenario in order to evaluate the protocol under these conditions.

The call rate (λc) chosen is 1 communication attempt every 180s (λc=0.005), that is, the user requests information from the installation with a mean value of 180s for 6000s, which is the simulation time used. This means that for one experiment, more than one route request is produced. It is necessary to clarify one point about this parameter; the route requests according to this parameter should be approximately 33, but the value is less than this as there will not be a new request while a session is in operation. Previous studies on the performance of AODV with Petri nets [24], [25] do only the search for one path every simulation. In the model presented, when the session time is finished another path can be requested by the source node. The session times used were 20s, 1 and 3 minutes, as explained in the section describing the scenario. The number of nodes used was 6 and the radio transmission range varied between 100m and 200m.

4.1 Lost and Recovered Paths

In Fig. 8 we show the performance of the AODV in the search, loss and recovery of a path to the destination. In A, we show the total percentage of paths found with respect to paths requested. It is clear that this percentage increases as radio range increases. With 100m, the percentage of possible paths is 37.65%, however, with 150m, 71.11% of the paths requested are possible, and if we increase the radio transmission range to 200m the percentage rises to 91.54%.

We also show the percentage of direct and indirect routes lost due to the movement of one of its nodes, see B and C in Fig. 8. Lost routes are lower when radio range is great as is to be expected. We can observe that multi hop paths suffered more losses. This is because more nodes participate in this type of communication; therefore the probability that one of the nodes moves is higher, leading to a higher probability of a break than in the case of a path formed by only two nodes, source and destination.

Finally, we show the percentage of direct and indirect routes recovered compared to the number of routes lost, D and E respectively. It was to be expected

that as radio range was increased, recovered routes increased correspondingly as D and E show. Given that the AODV gives preference to single-hop paths over multi-hop paths, the majority of recovered routes should be direct communications; but the results indicate that the majority of recovered routes were found through multi-hop communication. The reason is that in spite of the low probability of losing a path when radio range is high, if a direct route is lost, it is recoverd almost 100% of the time via a multi-hop path.

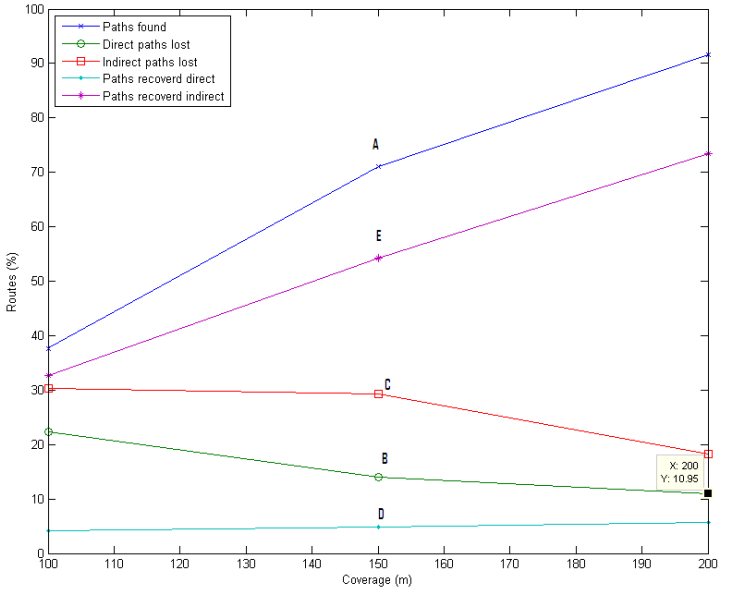


Fig. 8. Percentage of paths found with respect to path requests made$^{(A)}$, direct$^{(B)}$ or indirect$^{(C)}$ paths lost with respect to the number of established routes and recovered through single-hop$^{(D)}$ or multi-hop$^{(E)}$ communication when these had been lost

In Table. 1 labelled as "path lost" we show the sum of the percentages of direct and indirect routes lost with respect to those that were active. The results are shown for 3 radio ranges. Under the label "recovered paths", we show two percentages. The first (*) is the sum of the percentages of direct and indirect paths recovered compared with those lost. The second (**) shows the paths recovered compared to those found after a request which were in effect, active paths. The second percentage of "recovered paths" was obtained through the other two percentages in the table. Showing the same value in two ways anables us to look at the results from two different perspectives. The paths recovered with respect to those lost (*) enable us to see how the protocol reacts to a break of link and loss of path. The paths recovered compared to those active (**), enable us to understand in what measure the paths that are being used for transmission of information could have problems through a momentary loss of the path.

We can say that with a range of 100m, 52.68% of active paths are lost (the sum of direct and indirect losses, curves B and C in Fig. 8). Of the total of paths lost, 36.83% were recovered, which is equivalent to 19.4% of the paths that were found. We can also state that on 19.4% of all occasions, information was sent with some packets lost due to momentary losses of path, but it was possible to continue sending the information because the route was recovered within the time established by the AODV. Along the same lines, we can state that 47.32% (100% - 52.68%) of total communications were completed practically without problems. This is the percentage of paths not lost compared to active paths.

For a radio range of 150m in 43.24% of active paths some links were lost, causing a loss of path. 59.22% of the time, the lost links were recovered, which is equivalent to a recovery of 25.6% of active paths lost.

For a radio range of 200m, only 29.21% of active routes were lost, and of these paths 79.09% were recovered. That is to say, in 23.10% of active paths, there are problems on some occasion during the transmission, and in 6.11% (29.21%-23.10%) of active paths these problems could not be solved without loss of information.

Table 1. Percentage of paths lost and recovered for different radio transmission ranges

	Radio transmission range (m)		
	100	150	200
Path lost	52.68%	43.24%	29.21%
Recoverd paths			
(*) with respct to path lost	36.83%(*)	59.22%(*)	79.09%(*)
(**) with respect to active paths	19.4%(**)	25.6%(**)	23.10%(**)

4.2 Average Path Lifetime

The average path lifetime is the total time (the sum of the parts if there are breaks) at the end of the experiment during which there is a usable path between source and destination, divided by the number of different paths found. An example of this can be seen in Fig. 9 where we can see firstly the ideal path lifetime. This is the time for which the source and the destination can maintain communication, single-hop or multi-hop, until the source and the destination are definitively out of range.

Secondly, we can see the time that we want to measure. The path begins to be available when after a route request (RREQ), there is a route reply (RREP), and it is at this moment that the usable path lifetime begins. In both cases when the source and destination are out of range the time count stops and continues if there is a new path after the new search. The average path lifetime is obtained through the sum of the usable path lifetimes[9] divided by the number

[9] It is important to note that we modelled more than one path request during the simulation, and it is because of this that Fig. 9, showing the times, is repeated as many times as there are requests, and so we can define the average path lifetime as the sum of these times divided by the number of routes.

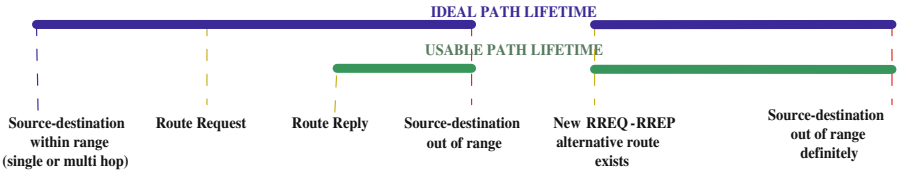


Fig. 9. Diagram of times showing the Path Lifetime

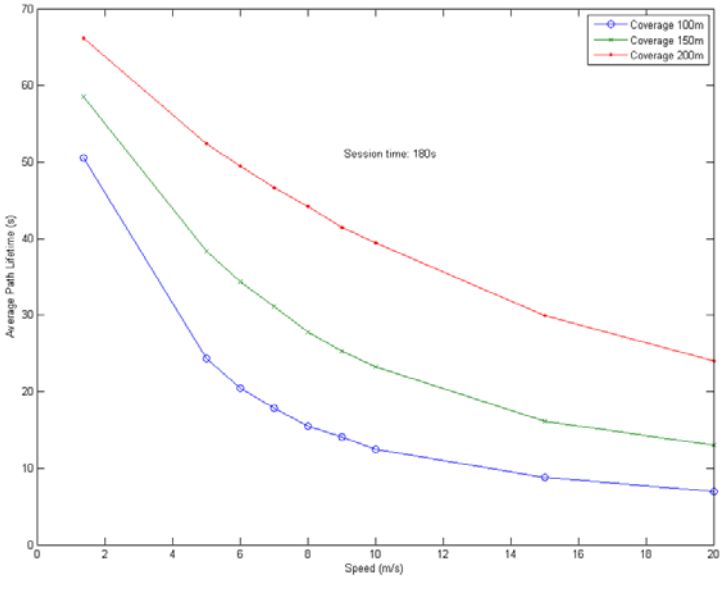


Fig. 10. Average path lifetime for a session time of 180s

of paths found throughout the simulation time. Average path times have been obtained for different session times. The performance is the same independent of the session time, but average path times values are not independent of this time. The difference in results is because the session time affects the period in which the paths active time is accumulating, and therefore affects the average path lifetime.

In Fig. 10, we can see the average path lifetime for a session time of 180s. It shows how the average time evolves as the speed of the nodes is increased from 1.38m/s to 20m/s, in function of the range. The average path lifetime decreases as the speed of the node increases. In turn, we can also see that as the radio transmission range is greater, the average time is also greater. The results obtained are comparable with those obtained by Ishibashi et al. (see Fig. 15 in [18]), in which they present the effects of mobility in an ad hoc network. Although the lifetime is only dependent on the mobility and transmission range, the density of the nodes in the network affects the quantity of links formed.

In [18] the authors use 50 nodes and the mobility model used is the random waypoint. Because of this, we cannot make a direct comparison between both sets of results, but it is possible to state that the basic performance is the same. The speed of the nodes affect the link lifetime.

5 Conclusions and Future Work

In this paper, formal models have been used to analyze the average path lifetime. These models represent an ad hoc network in which AODV is the routing protocol. The values obtained enable us to better understand the temporal performance of the paths created with this algorithm. It enables us to evaluate which services can be offered in an environment with the characteristics of the ad hoc network described here; where mobile users can request information in real time (images or alarms). Although in the experiments 3 radio ranges were used to give an overview of how this parameter affects maintenance of the path, we consider a radio transmission range of 150m to be the best choice as we have demonstrated in previous works [16], [17]. With this value of radio transmission range and taking into account the assumptions mentioned previously; the average path lifetime is 58.49s when the session time is 180s.

If we think of a scenario where the technicians are consulting images in order to control the normal operation of an industrial system, a session of 58s would be sufficient to observe the installation or how a process is working at a particular time. However, it is important to take into account that 43.24% of paths (Table. 1) are lost due to movement of the nodes, but that 59.22% (Table. 1) of these lost paths are recovered. Therefore, this type of network is best suited to offering images or alarm services, or to check the operation of a process within the plant at a particular moment. However, it is important to highlight that it would be difficult to offer services such as video streaming or voice. Providing multimedia applications in ad hoc networks is becoming a critical issue nowadays, but these applications are delay-sensitive and have high bandwidth requirements. Studies such as those made in [23] state that to provide efficient QoS routing over wireless ad hoc networks, problems such as scalability, power control, energy drain balancing and an efficient design of QoS MAC protocols need to be further investigated.

As we have stated in this paper, various suppositions were made in the development of the model, and therefore we should point out that the average path lifetime values would probably be a little lower. To address this, we are currently working on the introduction of traffic and transmission errors into the model in order to simulate the performance as close to reality as possible. Moreover, it is widely recognised that the performance of an ad hoc network varies according to the mobility model used [26], and because of this, we are also working on the use of the random waypoint mobility model, RW. With all of these improvements in the model, it will be possible to obtain values which are very close to those of real situations. This would enable study the repercussions of node mobility in the quality of service perceived by a user in supervision and control application within an industrial environment.

Acknowledgments. This work was supported by the MCyT (Spanish Ministry of Science and Technology) under the projects TSI2007-66637-C02-01/02, whose are partially funded by FEDER.

References

1. Molinero, F.G.: Real-Time Requirements of Media Control Applications. In: 19th Euromicro Conference on Real-Time Systems, Pisa, Italy (2007)
2. Silvestre, J., Sempere, V.: An architecture for flexible scheduling in Profibus Networks. Computer Standards & Interfaces 29, 546–560 (2007)
3. Perkins, C., Royer, E.: Ad-Hoc On-Demand Distance Vector Routing (AODV), RFC 3561 (July 2003)
4. Official IETF working group MANET webpage, http://www.ietf.org/html.charters/manet-charter.html
5. OLSR webpage, http://www.olsr.org
6. Eleonora, B., Marco, C., Franca, D., Luciana, P.: Lessons from an Ad Hoc Network test-bed: Middleware and routing issues. Ad Hoc & Sensor Wireless Networks 1, 125–157 (2005)
7. Eleonora, B.: Experimental Evaluation of Ad Hoc Routing Protocols. In: 3rd International Conference On Pervasive Computing and Communications Workshops (2005)
8. Gupta, A., Wormsbecker, I., Williamson, C.: Experimental evaluation of TCP performance in multi-hop wireless Ad hoc networks. In: IEEE Computer Society's 12th Annual International Symposium on Modelling, Analysis, and simulation of Computer and Telecommunications Systems (2004)
9. Tschudin, C., Gunningberg, P., Lundgren, H., Nordstrom, E.: Lessons from Experimental MANET Research. In: Conti, M., Gregori, E. (eds.) Ad Hoc Networks Journal, special issue on Ad Hoc Networking for Pervasive Systems
10. Conti, M., Giordano, S.: Multihop Ad Hoc Networking: The theory. IEEE Communications Magazine 45(4), 78–86 (2007)
11. Albero, T., Sempere, V., Mataix, J.: A study of mobility and reachability in Ad Hoc networks using stochastic activity networks. In: 2nd Euro NGI Conference: Next Generation Internet Desing and Engineering (2006)
12. Sanders, W.H.: UltraSAN users manual, University of Illinois (1995)
13. Clark, G., Courtney, T., Daly, D., et al.: The Mbius Modelling tool. In: 9th International Workshop on Petri Nets and Performance Models, pp. 241–250 (2001)
14. PERFORM Group (Performability Engineering Research Group) of Illinois at Urbana-Champaign, http://www.mobius.uiuc.edu/
15. Meyer, J.F., Movaghar, A., Sanders, W.H.: Stochastic Activity Networks: Structure, Behaviour and Application, In: International Conference On Timed Petri Nets, Turin, Italy, pp. 106–115 (1995)
16. Albero, T., Sempere, V., Mataix, J.: A study of multi-hop communications in Ad Hoc Networks using Stochastic Activity Networks. In: Workshop on Wireless and Mobility (2006)
17. Albero, T., Sempere, V., Mataix, J.: Estudio de alcanzabilidad en Redes Ad Hoc mediante redes de Actividad Estocstica. VI Jornadas de Ingeniera Telemtica (JITEL 2007), Mlaga (2007)
18. Ishibashi, B., Boutaba, R.: Topology and mobility considerations in mobile ad hoc networks. Ad Hoc Networks Journal, Elsevier 3(6), 62–776 (2004)

19. Maltz, D.A., Broch, J., Johnson, D.B.: Experiences designing and building a multihop wiereless Ad Hoc Network Testbed, Technical Report CMU-CS-99-116, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania (1999)
20. Giner, V.C., Escall, P.G., Oltra, J.M.: Modelling mobility tracking procedures in PCS systems using Stochastic Activity Networks. International Journal on Wireless Information Networks 9(4) (2002)
21. Lundgren, H., Lundberg, D., Nielsen, J., Nordstrom, E., Tschudin, C.: A large-scale testbed for reproducible ad hoc protocol evaluations. In: IEEE Wireless Communications and Networking Conference Record (2002)
22. Yongguang, Z., Wei, L.: An integrated environment for testing mobile ad-hoc networks (2002)
23. Xu, S., Saadawi, T.: Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks? IEEE Communications Magazine 39(6), 130–137 (2001)
24. Xiong, C., Murata, T., Tsai, J.: Networks using Colored Petri Nets. In: Kristensen, L.M., Billington, J. (eds.) Workshop on formal methods applied to defense systems, Adelaide, Australia. Conferences in Research and Practice in Information Technology, vol. 12 (2002)
25. Ma, H., Hu, Z., Wang, G.: A reliable routing algorithm in Mobile Ad Hoc Networks using Fuzzy Petri Net. In: Globecom Workshops. IEEE Communication Society (2004)
26. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. Wireless communications & Mobile Computing, special issue on Mobile Ad Hoc networking: research, trends and applications 2(5), 483–502 (2002)

Overall Delay in IEEE 802.16 with Contention-Based Random Access*

Sergey Andreev[1], Zsolt Saffer[2],
Andrey Turlikov[1], and Alexey Vinel[3]

[1] State University of Aerospace
Instrumentation (SUAI), Russia
serge.andreev@gmail.com
turlikov@vu.spb.ru
[2] Department of Telecommunications,
Budapest University of Technology,
and Economics (BUTE), Hungary
safferzs@hit.bme.hu
[3] St.Petersburg Institute for Informatics
and Automation of RAS (SPIIRAS), Russia
vinel@ieee.org

Abstract. In this paper we address the overall message delay analysis of IEEE 802.16 wireless metropolitan area network with contention-based multiple access of bandwidth requests. The overall delay consists of the reservation and scheduling components. Broadcast polling is used for bandwidth reservation with binary exponential backoff (BEB) collision resolution protocol and a simple scheduling is applied at the base station. An analytical model is developed with Poisson arrival flow for the Non Real-Time Polling Service (nrtPS) class. The model enables asymmetric traffic flows, different message sizes at the subscriber stations and also allows for Best Effort (BE) service class. An approximation of the mean overall delay is established for the nrtPS service class. The analytical model is verified by means of simulation.

Keywords: IEEE 802.16, WMAN, performance evaluation, bandwidth reservation, contention-based multiple access, BEB, queueing model.

1 Introduction

IEEE 802.16 is a notorious specification, which is recommended for Wireless Metropolitan Area Networks (WMANs). The standard specifies an air interface for Broadband Wireless Access (BWA) [1]. It proposes a high-speed access system supporting multimedia services and an extensive *quality-of-service* (QoS) guarantee. In IEEE 802.16 protocol stack the Medium Access Control (MAC) layer supports multiple Physical (PHY) layer specifications, each of them covering different operational environments.

* This work is partially supported by the NAPA-WINE FP7-ICT (http://www.napa-wine.eu) and the OTKA K61709 projects.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 89–102, 2009.

Many authors studied the performance of the various IEEE 802.16 features. In particular, the bandwidth requests mechanism to reserve a portion of the channel resources is frequently addressed. A detailed description of the reservation techniques and a general queueing model are given in the fundamental works [2] and [3]. The standard allows a *random multiple access* (RMA) reservation scheme and implements the truncated *binary exponential backoff* (BEB) protocol for the purposes of the collision resolution.

The asymptotic behavior of the BEB protocol was substantially addressed in the literature. In [4] it was shown that the BEB protocol is *unstable* in the infinitely-many users case. By contrast, [5] shows that the BEB is *stable* for any finite number of users, even if it is extremely large, and sufficiently low input rate. An exhaustive description of various analytical RMA models models may be found in [6] and [7]. The performance of the BEB algorithm in the framework of the reference RMA model ([8], [9]) is addressed in [10], which allowed a deeper insight into its operation. In the fundamental analysis of [11] an extremely useful Markovian model to analyze the performance of the BEB algorithm was first introduced.

Together with the analysis of the BEB itself, much attention is paid to its proper usage in IEEE 802.16 standard. It is known that the BEB algorithm may be adopted for both *broadcast* and *multicast* user polling. The efficiency of broadcast and multicast polling was extensively studied in [12] and [13]. Some practical aspects of the BEB application for the delay-sensitive traffic were considered in [14].

Considerable research effort is done also on overall performance aspects of the IEEE 802.16 system. For example in [15], [16] and [17] various frameworks are built and analyzed to guarantee a specified level of QoS. Furthermore, in [18] and [17] the overall system delay is estimated and verified. However none of these methods are dealing with overall delay in the context of contention-based random access.

In this paper we develop a first analytic approximation for the overall delay in the IEEE 802.16 system with broadcast polling.

The rest of the paper is structured as follows. Section II gives a brief overview of IEEE 802.16 MAC layer. In Section III we provide the description of the model and the notations. We conduct the overall delay analysis in Section IV. In Section V we verify the analytical results by means of simulation. Finally, we give our conclusion in Section VI.

2 Brief Overview of IEEE 802.16 MAC

IEEE 802.16 standard supports two operational modes: the mandatory *Point-to-MultiPoint* (PMP) and the optional mesh mode. In the centralized PMP architecture the *Base Station* (BS) is the main node. It is responsible for coordinating the communication process among the other nodes – *Subscriber Stations* (SSs). All communication among the SSs is directed through the BS and takes place on independent transmission channels of two types. In the *Downlink Channel* (DL) only the BS transmits data to the SSs, while in the *Uplink Channel* (UL) the

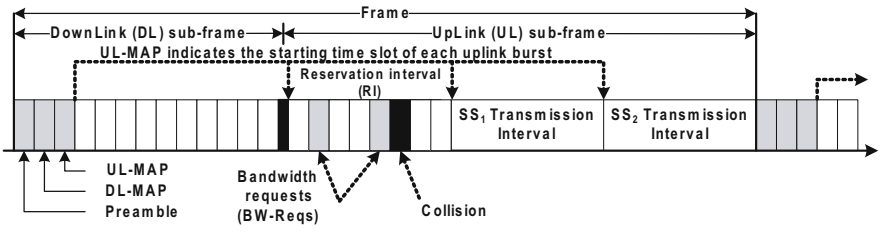


Fig. 1. IEEE 802.16 MAC frame structure in TDD/TDMA mode

data is sent by the SSs to the BS. Hence, there is no multiple access on the DL channel, while the UL channel is shared among multiple SSs.

The standard provides two channel allocation schemes: *Frequency Division Duplexing* (FDD) and *Time Division Duplexing* (TDD). In FDD the the DL and the UL channels are assigned to the different frequencies, while in TDD both channels are assigned to the same frequency, and are differentiated by assigning different time intervals to them. In this case the time is divided into fixed-length frames, which consist of the DL and the UL sub-frames corresponding to the DL and the UL channels, respectively. The length of the sub-frames can be varied dynamically. The SSs access the UL channel by means of *Time-Division Multiple Access* (TDMA).

The MAC frame structure can be seen in Figure 1. In the DL sub-frame the BS broadcasts data to all the SSs, and each of them captures only those addressed to it. Besides the DL scheduling, the BS is also responsible for the UL scheduling. The BS determines the number of slots to be allocated for each SS in the next UL sub-frame. This information is broadcasted in the UL-MAP message in the beginning of each frame. After receiving the UL-MAP message, the SS transmits data in the next UL sub-frame using the time slots which are granted to it.

The SS can initiate bandwidth reservation by sending a *Bandwidth Request* (BW-Req) message in the *Reservation Interval* (RI) in the beginning of each UL sub-frame. The standard defines contention-free polling mechanism (unicast) and contention-based random access polling mechanisms (multicast or broadcast) for bandwidth reservation. The duration of the RI is not specified by the standard explicitly. In case of contention-based random access, the defined collision resolution mechanism is the truncated *Binary Exponential Backoff* (BEB) protocol. Additionally, IEEE 802.16 enables piggybacking for sending BW-Reqs attached to data packets.

3 Model and Notations

3.1 Restrictions of the Model

Our model describes the IEEE 802.16 MAC with the following limitations:

R.1: The operational mode is PMP.

R.2: TDD/TDMA channel allocation scheme is used.

R.3: Messages of nrtPS and BE service classes are allowed, however we consider only the performance of the nrtPS service class.

R.4: The bandwidth reservation mechanisms is the contention-based broadcast polling.

R.4: The uplink scheduler applies a simple scheduling (see in 3.2).

R.6: One connection per SS is allowed.

R.7: Piggybacking is not used.

3.2 General Model and Scheduling

There are 1 BS and N SSs in the system, which together comprise $N+1$ stations. In this model we consider only the uplink traffic of messages. Each SS has infinite buffer capacity to store the waiting messages. Messages transmitted by the SSs consist of a number of data packets.

A BW-Req sent by a SS i represents the request for all i-messages, which are accumulated in its outgoing buffer since its last successful BW-Req sending.

For each SS the BS maintains an individual buffer with infinite capacity. At the end of each polling slot the BS performs an immediate processing of the successfully received BW-Req, if any, and of non-empty individual BS buffers of SSs, at which scheduling arises.

If BW-Req is received from SS i, then BS immediately assigns an individual request to each data message represented by the received BW-Req and these requests are put into the corresponding individual BS buffer of SS i (according to their order taken from BW-Req). If the individual BS buffer of SS i is empty upon receiving a BW-Req from that SS, then after putting the individual requests into the individual BS buffer of that SS the i-message corresponding to the first individual request is immediately scheduled for transmission on UL in the next frame. The frame duration is T_f. As exactly one i-message can be transmitted in one frame the BS schedules the i-messages corresponding to the waiting individual requests periodically after each T_f time until the individual BS buffer of SS i becomes empty.

The contention-based random access and the scheduling is illustrated in Figure 2.

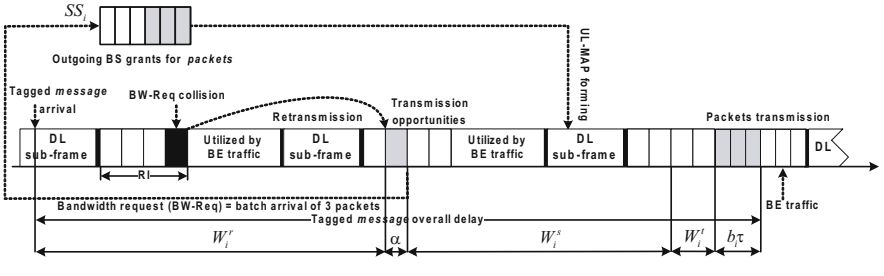


Fig. 2. Contention-based model and scheduling operation

This way one message will be scheduled from every non-empty individual BS buffers of SSs during processing a frame (its RI). This processing is repeated periodically in consecutive frames.

In the case when one or more SSs have no message of nrtPS service class to send on uplink, the system is allowed to utilize the unused uplink transmission capacity for uplink transmission of BE messages. This ensures a more efficient capacity utilizing. However the modeling of reservation and transmission of BE messages is out of scope of this paper.

3.3 Analytical Model

The message arrival process during each slot is Poisson at each SS. The duration of a transmission slot is τ. We express the number of arrivals in messages per time unit. The mean number of arriving i-messages per time unit is denoted by λ_i. Hence the overall arrival rate is $\lambda = \sum_{i=1}^{N} \lambda_i$. The messages are assumed to be of fixed length. Therefore, b_i denotes the size of an i-message, i.e. the number of packets (transmission slots) in a message arriving to SS i. The arrival processes and the message sizes (in transmission slots) at the different SSs are assumed to be mutually independent.

Denote the duration of the DL and UL sub-frames by T_d and T_u, respectively. T_{ri} stands for the duration of the RI and T_{ud} is the maximum available duration of the uplink data transmission in a frame. It follows that:

$$T_u = T_{ri} + T_{ud}.$$

The transmission time of a BW-Req is α. The reservation interval of each frame consists of K polling slots (transmission opportunities), whose sizes equal to the transmission time of a BW-Req. Hence $T_{ri} = K\alpha$ and we get:

$$T_{ud} = T_u - K\alpha. \tag{1}$$

Since in one frame exactly one i-message can be transmitted on uplink from every SSs, for the optimal capacity allocation of the SSs holds:

$$T_{ud} = \sum_{i=1}^{N} b_i. \tag{2}$$

3.4 Model Assumptions

We denote the *utilization* of SS i by ρ_i. Since each SS gets a chance to transmit on UL at most one message in each frame, we obtain for the utilization of SS i:

$$\rho_i = \lambda_i T_f. \tag{3}$$

Additionally, we formulate the following assumptions of our model:
A.1: The following relation holds for the arrival rate of each SS i:

$$\rho_i = \lambda_i T_f < 1, \quad i = 1, \ldots, N. \tag{4}$$

This relation ensures the stability of the model.
A.2: The time of BS processing including scheduling is negligible.
A.3: The channel propagation time is negligible.
A.4: The transmission channels are error-free.
A.5: BW-Req for i-messages arriving during RI can be sent first time in the RI of the next frame.

4 Overall Delay Analysis

The overall delay of an i-message arises mainly due to waiting of the i-message in the outgoing buffer of SS i to get access for successfully sending bandwidth request (waiting for reservation) and the corresponding queuing in the individual BS buffer of SS i (waiting for scheduling).

4.1 Overall Delay Definition

We define the *overall delay* (W_i) of the tagged i-message as the time interval spent from its arrival into the outgoing buffer of SS i up to the end of its successful transmission in the UL. It is composed of several parts:

$$W_i = W_i^r + \alpha + W_i^s + W_i^t + b_i\tau, \tag{5}$$

where W_i^r is the reservation delay, which is defined as the time interval from the i-message arrival to SS i until the start of successful transmission of the corresponding BW-Req to the BS.

α is the transmission time of a BW-Req.

We define the *scheduling time of the tagged i-message* as the the end of the polling slot, when the tagged i-message is scheduled by BS for transmission on UL in the next frame.

W_i^s is the scheduling delay, which is defined as the time interval from the end of sending a BW-Req of the tagged i-message to its scheduling time.

W_i^t is the transmission delay, which is defined as the time interval from the scheduling time of the tagged i-message to the start of its successful transmission in the UL sub-frame.

$b_i\tau$ is the transmission time of an i-message.

4.2 Reservation and Scheduling Delays

We consider the 2 most important terms of the overall delay (reservation and scheduling delays) together, since it results in a simpler queueing model as treating them separately.

Since SS i has an individual request buffer in BS and a fixed bandwidth for UL transmission in each frame assigned to it, the statistical behavior of a particular SS is independent of the behavior of the other SSs. Therefore the stochastic behavior of a particular SS can be modeled by an individual queueing model.

In the queueing model for the reservation and scheduling delays W_i^t does not need to be taken into account. Hence in this queueing model the service of the tagged i-message starts at its scheduling time, i.e. when the BS schedules that message for transmission on UL in the next frame. In case of empty individual BS buffer of SS i this happens at the end of successful BW-Req transmissions from that SS. Hence in this queueing model the busy periods can start only at the end of successful BW-Req transmissions from SS i. As SS i has fixed bandwidth for UL transmission in each frame assigned to it, the service time is T_f. Thus the appropriate model is an M/D/1 queueing model, in which the service time equals T_f. Furthermore we observe that the service of the arriving i-message can not start until the next successful BW-Req transmissions from SS i even if the individual BS buffer of SS i is empty. Although this is a vacation-like property, we rather apply the approach of [9] by means of the residual service time, since it does not need any higher moments and hence it is simpler.

Applying the mean delay formula of the approach of the residual service time in our model with the corresponding parameters leads to

$$E\left[W_i^r + W_i^s\right] = E\left[W_i^0\right] + \frac{\lambda_i T_f^2}{2(1-\lambda_i T_f)}, \tag{6}$$

where W_i^0 is the *initial message delay*, which is the sum of the reservation and scheduling delays conditioning on the fact that the arriving i-message sees the system empty.

We remark here that (6) is an approximation. The approach of the residual service time – exactly as the vacation model approach with exhaustive service – assumes that the service is work conserving as far as there are i-messages in the system. However if there are i-messages waiting for reservation when the individual BS buffer of SS i becomes empty then the principle of work conserving does not hold any more for this model, because the service stops.

4.3 Initial Message Delay – In General

We assume that in stationary situation the successful BW-Req transmission at SS i in a polling slot has a constant probability, $p_i^{st} > 0$.

In the following we introduce several quantities in order to determine $E\left[W_i^0\right]$ in our model. We define W_i^{rs} as the time interval from the begin of first try of sending a BW-Req of the tagged i-message until the start of successful transmission of the corresponding BW-Req to the BS. Due to the constant probability of the successful BW-Req transmission at SS i in a polling slot W_i^{rs} is geometric in terms of the number of polling slots. More precisely its distribution is given as:

$$P\left\{W_i^{rs} = \lfloor \frac{n}{K} \rfloor T_f + (\frac{n}{K})^* K\alpha\right\} = (1-p_i^{st})^n p_i^{st}, \quad n \geq 0, \tag{7}$$

where $\lfloor c \rfloor$ and $(c)^*$ stand for the integral part and the fractional part of c, respectively.

By definition W_i^{rb} is the interval seen by a first arriving i-message after a successful BW-Req transmission until the begin of first try of sending the BW-Req from that SS. Due to the empty system condition W_i^0 is given as

$$W_i^0 = W_i^{rb} + W_i^{rs}. \tag{8}$$

After a successful BW-Req transmissions until the begin of first try of sending a BW-Req from SS i all arrivals occurs only in the last T_f part. Hence

$$E\left[W_i^{rb}\right] = \frac{T_f}{2}. \tag{9}$$

It is shown in the Appendix that

$$E\left[W_i^{rs}\right] = \frac{\left(1-p_i^{st}\right)^K}{1-\left(1-p_i^{st}\right)^K} T_f + \frac{\left(1 - p_i^{st} K \left(1-p_i^{st}\right)^{K-1} - \left(1-p_i^{st}\right)^K\right) \frac{1-p_i^{st}}{p_i^{st}}}{1-\left(1-p_i^{st}\right)^K} \alpha. \tag{10}$$

Applying (9) and (10) in (8) leads to

$$\begin{aligned} E\left[W_i^0\right] = {} & \frac{T_f}{2} + \frac{\left(1-p_i^{st}\right)^K}{1-\left(1-p_i^{st}\right)^K} T_f \\ & + \frac{\left(1 - p_i^{st} K \left(1-p_i^{st}\right)^{K-1} - \left(1-p_i^{st}\right)^K\right) \frac{1-p_i^{st}}{p_i^{st}}}{1-\left(1-p_i^{st}\right)^K} \alpha. \end{aligned} \tag{11}$$

For high traffic load p_i^{st} can be approximately determined by means of the independent conditional collision probability assumption proposed by Bianchi [11], which leads to a nonlinear equation (see also [13]).

However in other traffic ranges it does not hold, since during the collision resolution process the SSs influences each other. Thus in general the determination of p_i^{st} is a difficult task. Therefore, as a first analytic approximation, we consider a simplified symmetric model to determine $E\left[W_i^0\right]$.

4.4 Initial Message Delay – Symmetric System

Let K per frame equal to 1 and we set the message sizes of all SSs b_i equal to 1 packet. Further we set all λ_i values equal, which makes the system symmetric in terms of the message arrival flows.

The performance of the BEB collision resolution protocol should be optimized for the considered system settings. In [13] after the extensive analysis of the BEB operation it was established that the optimal value of the BEB parameter W

(initial contention window) should be equal to $2N - K$, where N – number of the nrtPS SSs in the system. We remark here that the polling slots of W can be distributed over more frames. The second BEB parameter m (backoff stage) should be equal to 0 for the optimal BEB protocol.

Therefore, the optimized BEB is reduced to the Aloha protocol [13], where each backlogged SS (the one that has at least one message ready for transmission) chooses one of W polling slots following the message arrival uniformly. In case of collision the SS repeats the choice of a random polling slot to retransmit its BW-Req until the transmission is finally successful. Once BW-Req is successfully transmitted to the BS in a polling slot, the queue of messages that belong to the corresponding SS is updated at the BS. Therefore, the information of all the messages accumulated during the contention process is transfered to the BS and the service starts.

In order to establish the initial message delay of a tagged SS i we consider the sequence of service times, i.e. the ends of the polling slots in the frame following the one where a BW-Req was transmitted successfully. We conclude, that as arrival flow is Poisson, the newly arrived message firstly waits for $\frac{T_f}{2}$ time before the first transmission attempt of a BW-Req according to the PASTA property (for a more thorough explanation see [19]). Then the contention process starts, which adds to the initial delay some random number of frames. Below we give an estimation on this random number.

We consider the following simple linear feedback model. Notice that in terms of bandwidth requesting each SS may be either *thinking* or *backlogged*. The thinking SS has no message ready for transmission and generates one during a frame with the probability $z = 1 - e^{-\frac{\lambda}{N}Tf}$. Once a new message is generated, the SS enters the backlogged state where no new arrivals are possible. This corresponds to the real system where after the first arrival an SS starts the contention process and the subsequent arrivals are irrelevant to establish the sought contention delay. Once the transmission of a BW-Req is successful, the SS enters the thinking state and is able to generate new messages. In each polling slot the backlogged SS attempts to transmit a BW-Req with the probability $p = \frac{2}{W+1}$.

We describe the considered linear feedback model [20] with a Markov chain consisting of $N + 1$ states (see Figure 3). Each state corresponds to the number of the backlogged SSs in the system at the moment of the service time $N^{(t)}$. The transition probabilities for the considered Markov chain are as follows:

$$p_{ij} = \Pr\{N^{(t+1)} = j | N^{(t)} = i\} = \tag{12}$$

$$= \begin{cases} 0, \ j \leq i-2 \\ ip(1-p)^{i-1}(1-z)^{N-i+1}, \ j = i-1 \\ ip(1-p)^{i-1}(N-i+1)z(1-z)^{N-i} + \\ +(1-ip(1-p)^{i-1})(1-z)^{N-i}, \ j = i \\ ip(1-p)^{i-1}\binom{N-i+1}{j-i+1} z^{j-i+1}(1-z)^{N-j} + \\ +(1-ip(1-p)^{i-1})\binom{N-i}{j-i} z^{j-i}(1-z)^{N-j}, \ j \geq i+1. \end{cases}$$

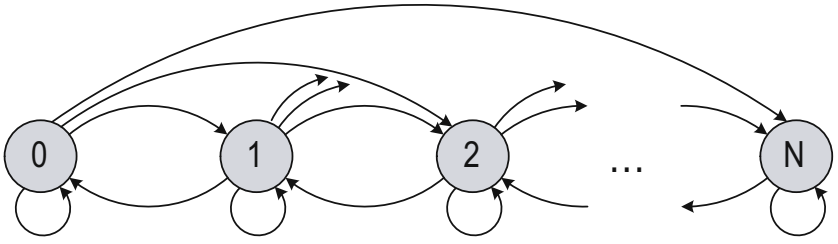


Fig. 3. Markov chain for linear feedback model

It may be shown that the considered chain is finite and irreducible for $p, z > 0$. Therefore, a stationary probability distribution always exists. This distribution may be obtained, for instance, by solving a system of $N + 1$ linear equations:

$$\begin{cases} P_j = \sum\limits_{i=0}^{N} P_i p_{ij}, \ j = 0, 1, \ldots, N \\ \sum\limits_{i=0}^{N} P_i = 1. \end{cases} \tag{13}$$

Using the stationary probability distribution one may obtain the average number of the backlogged SSs

$$B = \sum_{n=1}^{N} nP_n \tag{14}$$

and the stationary success probability

$$S = \sum_{n=0}^{N} s(n, p)P_n, \tag{15}$$

where $s(n, p) = np(1 - p)^{n-1}$. Finally, the mean delay in the considered linear feedback model is given by the Little's result, that is $D = \frac{B}{S}$.

Combining the above, the initial message delay in the symmetric system is given by the following expression:

$$E\left[W_i^0\right] = T_f(D + \frac{1}{2}). \tag{16}$$

4.5 Transmission Delay – Symmetric System

Remember, that each SS has a fixed position in the uplink subframe, that is, the transmission delay of SS i is $(i - 1)\tau$. Summarizing it over every SSs yields the transmission delay under symmetric settings as

$$W_i^t = \frac{1}{N} \sum_{i=1}^{N} (i - 1)\tau = \tau \frac{N - 1}{2}. \tag{17}$$

4.6 Mean Overall Message Delay

Applying (5) in symmetric system, the mean overall message delay is given as:

$$E\left[W_i\right] = E[W_i^r + W_i^s] + \alpha + E\left[W_i^t\right] + \tau. \tag{18}$$

Accounting for (18), (6), (16) and (17) the mean overall message delay for the symmetric system can be expressed by:

$$E\left[W\right] = T_f(D + \frac{1}{2}) + \frac{\frac{\lambda}{N}T_f^2}{2(1 - \frac{\lambda T_f}{N})} + \tau\frac{N+1}{2} + \alpha. \tag{19}$$

5 Simulation Results

In order to validate the considered analytical model a simulation program for IEEE 802.16 MAC was developed. The program is a time-driven simulator that accounts for the discussed restrictions on the considered system model. The applied simulation parameters of IEEE 802.16 MAC and PHY, which follows [21], are summarized in Table 1.

Table 1. Basic IEEE 802.16 simulation parameters

Parameter	Value
PHY layer	OFDM
Frame duration (T_f)	5 ms
DL/UL ratio	50:50
Channel bandwidth	7 MHz
MCS	16 QAM $^3/_4$
Packet length	512 $Byte$
BW-Req duration (α)	0.17 ms

For the purposes of simplicity we again restrict our practical explorations to the symmetric system case. This enables a better visibility of the below comparison results.

In Figure 4 we conduct the comparison of the overall delay for the practical system and the established theoretical estimation. We set the number of nrtPS SSs in the system N equal to 6 and run each simulation point for approximately 10 minutes of the real time. Firstly, we notice that the derived analytical expression gives a good estimation on the realistic overall delay value.

Another interesting observation is the intricate shape of the theoretical overall delay curve. Remember, that the analytic formulas are approximations, since the work conserving does not hold any more for this model if there are i-messages waiting for reservation when the individual BS buffer of SS i becomes empty. Therefore the i-messages actually waiting for reservation experience an additional waiting time compared to the work conserving case, where the service

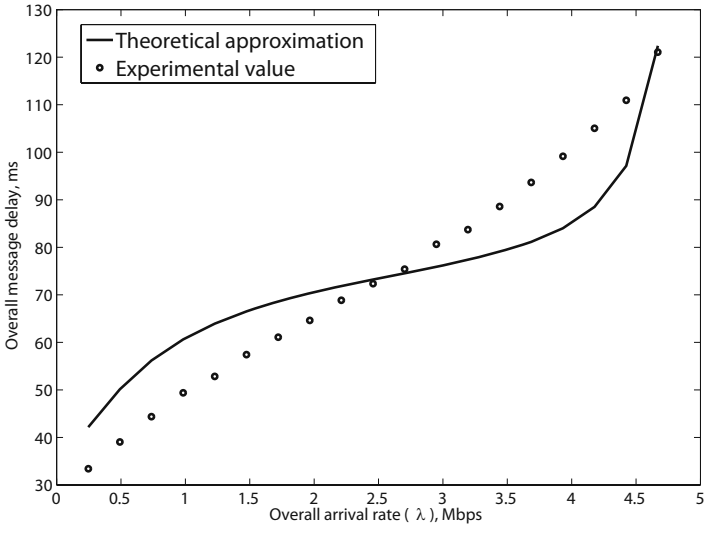


Fig. 4. Verification of the derived model for the symmetric system case

would continue. This supplementary waiting time of the tagged i-message takes until the next successful BW-Req transmission from SS i. It follows that our approach with the work conserving assumptions underestimates the real waiting time. On the other hand, the analytical approach overestimates the reservation time since the first message in a batch transmitted in one BW-Req waits longer than the others. Therefore we see that the shape of the curve changes. However the result gives a good approximation for the practical value.

6 Conclusion

In this paper we have developed an analytical approximation for the mean overall message delay of nrtPS traffic in IEEE 802.16 wireless network. This model accounts for both reservation delay and scheduling delay components and enables asymmetric Poisson arrival flows and different message sizes. More importantly, the contention-based broadcast polling of subscriber stations is the studied bandwidth reservation mechanism.

For symmetric system a simple linear feedback model is used to estimate the contention delay. As our experiments show, the established theoretical overall delay gives a good approximation for the practical values obtained with simulation.

The analytical approach used for symmetric system (the linear feedback model) can be extended for the asymmetric system as well, where arrival flows and/or message sizes are not equal. Moreover, the model may be also extended to account the case of $K > 1$ and the various BEB parameters.

Acknowledgement

Dr. Alexey Vinel acknowledges the support of Alexander von Humboldt Foundation for the funding of this work, which was done in part during his stay in Germany.

References

1. IEEE Std 802.16e-2005, Piscataway, NJ, USA (December 2005)
2. Rubin, I.: Access-control disciplines for multi-access communication channels: Reservation and tdma schemes. IEEE Transactions on Information Theory 25(5), 516–536 (1979)
3. Boxma, O.J., Groenendijk, W.M.: Waiting times in discrete-time cyclic-service systems. IEEE Trans. on Comm. 36(2), 164–170 (1988)
4. Aldous, D.: Ultimate instability of exponential back-off protocol for acknowledgment based transmission control of random access communication channels. IEEE Transactions on Information Theory 33(2), 219–233 (1987)
5. Goodman, J., Greenberg, A., Madras, N., March, P.: Stability of binary exponential backoff. Journal of the ACM 35(3), 579–602 (1988)
6. Chlebus, B.: Randomized Communication in Radio Networks. In: Pardalos, P., Rajasekaran, S., Reif, J., Rolim, J. (eds.) Handbook of Randomized Computing, vol. 1, pp. 401–456 (2001)
7. Tsybakov, B.: Survey of ussr contributions to random multiple-access communications. IEEE Transactions on Information Theory 31(2), 143–165 (1985)
8. Tsybakov, B., Mikhailov, V.: Free synchronous packet access in a broadcast channel with feedback. Problems of Information Transmission 14(4), 259–280 (1978)
9. Bertsekas, D., Gallager, R.: Data Networks. Prentice-Hall, Englewood Cliffs (1992)
10. Song, N., Kwak, B., Miller, L.: On the stability of exponential backoff. Journal Research of NIST 108, 289–297 (2003)
11. Bianchi, G.: Performance analysis of the ieee 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications 18(3), 535–547 (2000)
12. Lin, L., Jia, W., Lu, W.: Performance analysis of ieee 802.16 multicast and broadcast polling based bandwidth request. In: IEEE Wireless Communications and Networking Conference, vol. 1, pp. 1854–1859 (2007)
13. Andreev, S., Turlikov, A., Vinel, A.: Contention-based polling efficiency in broadband wireless networks. In: International Conference on Analytical and Stochastic Modelling Techniques and Applications, vol. 1, pp. 295–309 (2008)
14. Alanen, O.: Multicast polling and efficient voip connections in ieee 802.16 networks. In: 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, vol. 1, pp. 289–295 (2007)
15. Paschos, G.S., Papapanagiotou, I., Argyropoulos, C.G., Kotsopoulos, S.A.: A heuristic strategy for ieee 802.16 wimax scheduler for quality of service. In: 45th Congress FITCE (2006)
16. de Moraes, L.F.M., Maciel, P.D.: A variable priorities mac protocol for broadband wireless access with improved channel utilization among stations. In: Int. Telecomm. Symp., vol. 1, pp. 398–403 (2006)
17. Chang, Y.-J., Chien, F.-T., Kuo, C.-C.J.: Delay analysis and comparison of ofdm-tdma and ofdma under ieee 802.16 qos framework. In: IEEE Global Telecomm. Conf (GLOBECOM), vol. 1, pp. 1–6 (2006)

18. Iyengar, R., Iyer, P., Sikdar, B.: Delay analysis of 802.16 based last mile wireless networks. IEEE Global Telecommunications Conference 5, 3117–3127 (2005)
19. Saffer, Z.s., Andreev, S.: Delay analysis of ieee 802.16 wireless metropolitan area network. In: Int. Workshop on Multiple Access Communications (MACOM) (2008)
20. Kleinrock, L.: Queueing Systems: Volume II – Computer Applications. Wiley Interscience, Hoboken (1976)
21. Sivchenko, D., Bayer, N., Xu, B., Rakocevic, V., Habermann, J.: Internet traffic performance in ieee 802.16 networks. In: European Wireless (2006)

A Mean of W_i^{rs}

Using (10) the mean of W_i^{rs} can be expressed for $0 < p_i^{st} < 1$ as

$$E\left[W_i^{rs}\right] = \sum_{j=0}^{\infty} \left(jT_f \sum_{n=jK}^{(j+1)K-1} \left(1-p_i^{st}\right)^n p_i^{st} + \sum_{n=jK}^{(j+1)K-1} (n-jK)\,\alpha \left(1-p_i^{st}\right)^n p_i^{st} \right). \quad (20)$$

Rearranging results in

$$\begin{aligned}
E\left[W_i^{rs}\right] &= \sum_{j=0}^{\infty} \left(jT_f \left(1-p_i^{st}\right)^{jK} \sum_{n-jK=0}^{K-1} \left(1-p_i^{st}\right)^{n-jK} p_i^{st} \right. \\
&\quad \left. + \alpha p_i^{st} \left(1-p_i^{st}\right)^{jK} \sum_{n-jK=0}^{K-1} (n-jK) \left(1-p_i^{st}\right)^{n-jK} \right) \quad (21) \\
&= \sum_{j=0}^{\infty} \left(jT_f \left(1-p_i^{st}\right)^{jK} \left(1-\left(1-p_i^{st}\right)^K\right) \right. \\
&\quad \left. + \alpha \left(1-p_i^{st} K \left(1-p_i^{st}\right)^{K-1} - \left(1-p_i^{st}\right)^K\right) \frac{1-p_i^{st}}{p_i^{st}} \left(1-p_i^{st}\right)^{jK} \right) \\
&= T_f \left(1-\left(1-p_i^{st}\right)^K\right) \sum_{j=0}^{\infty} j \left(\left(1-p_i^{st}\right)^K\right)^j \\
&\quad + \alpha \left(1-p_i^{st} K \left(1-p_i^{st}\right)^{K-1} - \left(1-p_i^{st}\right)^K\right) \frac{1-p_i^{st}}{p_i^{st}} \sum_{j=0}^{\infty} \left(\left(1-p_i^{st}\right)^K\right)^j \\
&= T_f \left(1-\left(1-p_i^{st}\right)^K\right) \frac{\left(1-p_i^{st}\right)^K}{\left(1-\left(1-p_i^{st}\right)^K\right)^2} \\
&\quad + \alpha \left(1-p_i^{st} K \left(1-p_i^{st}\right)^{K-1} - \left(1-p_i^{st}\right)^K\right) \frac{1-p_i^{st}}{p_i^{st}} \frac{1}{1-\left(1-p_i^{st}\right)^K} \\
&= \frac{\left(1-p_i^{st}\right)^K}{1-\left(1-p_i^{st}\right)^K} T_f + \frac{\left(1-p_i^{st} K \left(1-p_i^{st}\right)^{K-1} - \left(1-p_i^{st}\right)^K\right) \frac{1-p_i^{st}}{p_i^{st}}}{1-\left(1-p_i^{st}\right)^K} \alpha.
\end{aligned}$$

We remark here that $p_i^{st} = 1$ implies $W_i^{rs} = 0$ and thus (21) holds also for $p_i^{st} = 1$.

Analyzing the Impact of Various Modulation and Coding Schemes on the MAC Layer of IEEE 802.11 WLANs

Osama M.F. Abu-Sharkh[1] and Ahmed H. Tewfik[2]

[1] Department of Communications Engineering
Princess Sumaya University For Technology
osama@psut.edu.jo
[2] Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis
tewfik@umn.edu

Abstract. The throughput of the medium access control sub-layer in IEEE 802.11 wireless local area network depends on the performance of the network at the physical layer level. In this paper, we perform cross layer analysis between the medium access control and the physical layers in order to study the behavior of the network including the achieved throughput for various types of modulation and coding schemes. In our analysis, we take into account the packet error rate of the schemes as a loss factor in an improved Markov Chain model. The model is in consistent with the DCF access mechanism of IEEE 802.11 standard and it includes all of its parameters in different operating conditions. Expressions for throughput and average service time of packets are provided. The analytical expressions are solved using MATLAB and the model is validated by experiments.

Keywords: 802.11, Markov Chain, Modeling.

1 Introduction

Growth of wireless packet data applications drives the rapid evolution of next generation wireless networks. Accurate analysis should be conducted on the current wireless standards to obtain precise picture of what steps should be taken to eliminate any deficiencies and make accurate improvements. One of the adopted techniques for studying the behavior of the network in recent years is the cross-layer-based analysis; whereby values in different layers of a subsystem are linked together and co-analyzed. In a IEEE 802.11 wireless network [1], the medium access control (MAC) and physical (PHY) layers can be co-analyzed to obtain actual and accurate values of the achieved throughput by different devices connected to the network.

The MAC throughput is strongly dependant on the following factors. First, the protocol timing overheads such as interframe spacing and the time of acknowledgements. Second, the time spent in the random backoff counter where the value range increases exponentially with transmission failures. It also depends on the transmission time of other users connected to the same network and sharing the same medium. Finally, it

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 103–115, 2009.

depends on the packet error rate (PER) at the physical layer and the transmission duration of each packet. Since the PER strongly depends on the modulation and coding schemes at the physical layer, the MAC throughput implicitly depends on the used scheme.

In [2], we introduced a Markov Chain model for the DCF access mechanism to precisely analyze the effect of the first three factors on the MAC throughput. The model was validated experimentally and by simulations. We also studied in depth the effect of the transmission duration for each packet on the system. We found deficiencies in the performance of the network when stations emit data at different rates.

The main contributions in this paper are summarized as follows. We analyze the effect of various types of modulation and coding schemes at the physical layer level on the MAC throughput. The studied schemes are Uncoded QPSK, Barker, CCK and PBCC which are used in 802.11b wireless LANs. We conduct the analysis by expanding our introduced MAC model to a cross-layer-based model. We include in our analysis, the PER and SNR based on an in depth study of the IEEE 802.11 physical layer introduced in [3]. We use appropriate values of PER and SNR according to the transmission speed of stations.

The rest of the paper is organized as follows. In Section II, we review relevant work in the literature. In Section III, we describe the cross-layer-based Markov Chain model. We provide expressions for the throughput and the average service time in Section IV. Finally, we discuss the numerical results in Section V.

2 Relevant Work

Relevant work is summarized in the following. In [4], Bianchi introduced a Markov chain model to compute the 802.11 DCF throughput. He made many assumptions in order to simplify the analysis. Many enhancements were then made on Bianchi's model to make it more consistent with the standard. Most studies performed analysis on the MAC layer and assumed ideal channel conditions. Recent studies considered the addition of the physical layer parameters and their effect on the network performance. In [5], the authors proposed an analytical model that calculates the performance of the standard taking into account the transmission errors for the IEEE 802.11a protocol. In addition to the collision between packets, they added the transmission errors in calculating the probability of packet loss. They assumed in their calculations that both the transmission errors and the collisions are independent. In [6], Helkov and Spasenovaski analytically analyzed the impact of an error-prone channel overall performance. They had a similar approach of the previous study in calculating the probability of packet loss. Finally, Manshaei et al proposed in [7], an analytical model that accounts for the positions of stations with respect to the access point while evaluating the performance of the MAC layer. They showed that the saturation throughput per station is strongly dependant not only on the station's position but also on the positions of other stations.

In [5] and [6], the authors included the PER without taking the channel aspects and its operating conditions in their calculations. They also did not show in their analysis, the dependency of PER on the SNR. On the other hand, although expressions for PER was analytically derived in [7], many assumptions and approximations were made to

simplify the analysis. For example, the authors assumed a simple path loss model that only considers the attenuation of power caused by the distance between the emitting terminal and the access point.

In [8], we used Markov Chains in a cross layer environment to model the IEEE 802.11 DCF access mechanism of the MAC layer for systems with multiple antennas. We studied the impact of adding MIMO links to achieve spatial multiplexing using ZF and MMSE on the performance of the MAC. All the analysis were conducted in a single rate environment and all stations were assumed to transmit data at the same speed.

3 The Cross-Layer-Based Models

Our proposed station model is shown in Fig. 1. In this paper we call stations which transmit data at full rate as fast stations and the ones which transmit data at lower rates as slow stations. Note that we model fast and slow stations in a similar manner. We use fast stations in deriving all expressions. Slow stations satisfy similar expressions with different values unless otherwise mentioned.

The model follows the operation of the IEEE 802.11 DCF. We describe it in the following.

Backoff states

Before a station starts transmission, it senses the channel to determine if it is idle. If the channel is idle for a DIFS, the station starts sending on the channel. Otherwise the station defers transmission for a backoff period of time β that is determined by randomly choosing an interval within its contention window. It does so by setting a backoff counter to the value β and decrementing it progressively. The backoff counter is stopped when a transmission is detected on the channel and decremented when the channel is sensed idle again. The size of the contention window is increased to double the previous size for every unsuccessful transmission until it reaches its maximum value. The packet is dropped from the queue after M unsuccessful attempts.

In our model, let a station be in one of the backoff states B_{ij}. If the channel is idle during the last time slot, the station decrements its backoff timer and enter B_{ij-1} or else it stays in B_{ij}. We denote the minimum and the maximum values of the contention window by W and W_{max} respectively. The backoff counter V_i is selected uniformly from [l, 2^i W] and the packet is retransmitted until the station reaches the backoff stage i such that $2^m W = W_{max}$. From that point on, the backoff counters is always chosen in the range [l, W_{max}]. Furthermore, if the packet is lost, it will be retransmitted until the total number of transmission attempts equal to the maximum number of retry limits M.

Note that we included the DIFS period of time that the station waits after sensing an idle medium in the channel model. This is discussed in the following section.

Transmission states

The station starts transmitting on the channel when it reaches the first state of any ith backoff stage. An ACK (CTS) frame is sent by the receiver upon successful reception of the transmitted frame. The station waits a period of time r that is equal to ACKTimeout

(CTSTimeout) in the standard before it detects that its previous transmission was not received successfully. During that time, the station is in the AT_{ij} states. If no ACK (CTS) is sent, the station enters the unsuccessful transmission states U_{ij}.

The station enters successful transmission states S_i, if the data frame was not lost due to collisions or errors in the channel. Successful transmission is completed when the station receives the ACK. During that time, it is in the AK_i states.

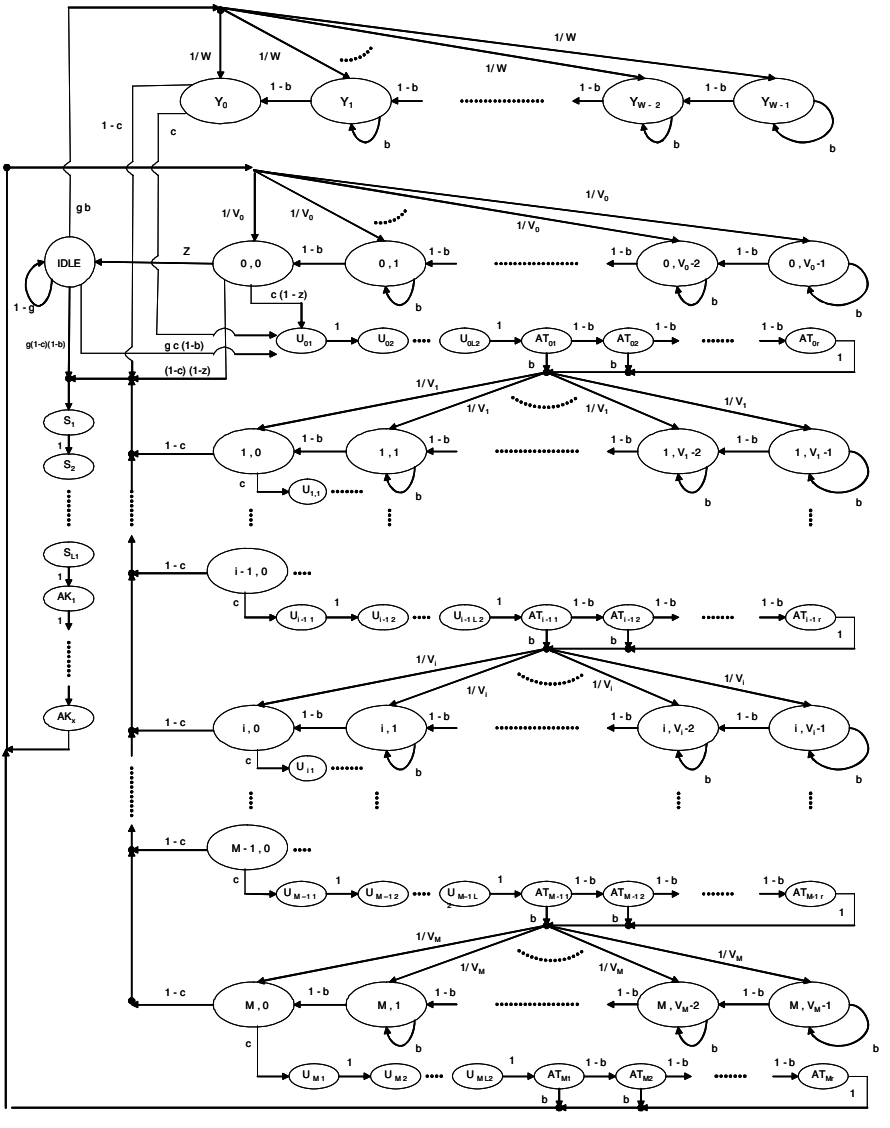


Fig. 1. The Station Model

Note that after any successful transmission, it is mandatory in the standard for the station to enter the first backoff stage even if it has a pending frame in its queue as shown in Fig. 1.

Idle states

The station enters state D only if its transmission queue is empty and its backoff timer is zero. It leaves D when it gets at least one frame from the upper layers. If the medium is idle, the station initiates frame transmission on the channel. This frame will be successfully transmitted when the frame is not lost due to a collision or channel errors. If the medium was busy, the stations enters Y_i and starts its backoff procedure with a minimum contention window. When the backoff timer reaches zero, the station initiates frame transmission with probability 1 given that there is at least one frame in its queue.

Basic and RTS/CTS access mechanisms

Since the basic or RTS/CTS access mechanisms can be employed in the model, we use different notations for the frame lengths depending on where the frame is used. We denote the length of the transmitted frame by $L1_f$ when successful transmission occurs. This frame length is only equal to the length of a data frame. On the other hand, we denote the frame length by $L2_f$ when the frame is lost. Frame length $L2_f$ is equal to either, the length of a data frame or an RTS frame.

Note that the station goes from the current state to itself, or another state, every time slot; i.e. the model has a constant transition time that equals the aSlotTime time interval of the standard. Therefore, we normalize the time duration of all variables to be a multiple number of time slots. For example, to calculate the length of data frame $L1_f$ of a station, we actually count the number of time slots it takes a frame to be transmitted on the channel. Thus, we calculate $L1_f$ as

$$L1_f = \frac{TheLengthInBits}{aSlotTime.DataRate_f}. \quad (1)$$

The Number of time slots needed to transmit a frame depends on the transmission rate of the station. For example, it takes 400 time slots to transmit a frame whose size is 1000 bytes when the data rate is 1 Mbps, and it takes only 37 time slots to transmit the same frame when the data rate is 11 Mbps.

Since the channel impacts the operation of all stations, we use a separate model for the channel.

The channel model is shown in Fig. 2. The channel is in the idle state E when there is no transmission, and it is busy when it is in one of the following states,

- successfully transmitting a frame of a fast station O_{fi}, (i=, 1,2,..$L1_f$),
- successfully transmitting a frame of a slow station O_{fsi}, (i=, 1,2,..$L1_s$),
- completing the frame exchange sequence of a successful transmission for a fast station OK_{fi}, (i=, 1,2,..x),
- completing the frame exchange sequence of a successful transmission for a slow station OK_{si}, (i=, 1,2,..x),

- transmitting two or more collided frames from fast stations $N_{f\,f\,i,}$ (i=, 1,2,..$L2_f$ +DIFS),
- transmitting two or more collided frames from slow stations $N_{ssi,}$ (i=, 1,2,..$L2_s$ +DIFS),
- transmitting two or more collided frames from a mix of fast and slow stations N_{fsi}, (i= 1,2,.. L2+DIFS; L2=max($L2_f$, $L2_S$)).

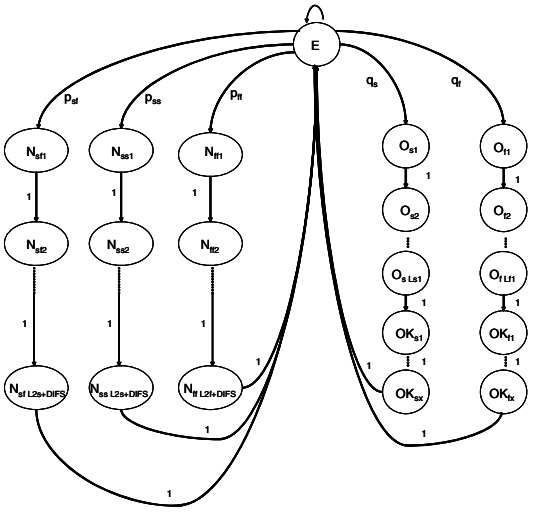


Fig. 2. The Channel Model

Note that x represents the period of time the channel is busy after a successful transmission. It is equal to a time duration of (SIFS + δ + ACK + δ + DIFS) for the basic access mechanism, or (RTS + δ + SIFS + CTS + δ + SIFS + δ + SIFS + ACK + δ + DIFS) for RTS/CTS, where δ is the propagation delay.

In the DCF, the backoff counter of a station is decremented in each idle time slot, frozen during channel activity periods and resumed after the medium is sensed idle again for a DIFS. The station resumes the backoff counter to the discrete value it had at the instant of time the busy channel period started. For example, suppose the backoff counter is decremented to 3 during an idle slot. Then this value is frozen during the busy channel period and resumed, again to value 3, only a DIFS after the end of the busy period. As a consequence, it is decremented to value 2 only a time slot after the DIFS. This happens when the station transits from state B_{ij} to $B_{i\,j-1}$.

Based on our models, we derive expressions for throughput in the following section.

4 Throughput

Throughput of a station by definition is the volume of data that the station successfully transmits. Therefore, we have to first find the probability that a station is successfully sending a frame to find the throughput. The computation of this probability

is equivalent to calculating P(Si), the probability of being in state Si, a successful transmission state, for all i = 1 to $L1_f$ from the station model.

From Fig.1, all the probabilities of being in the successful transmission states are equal, i.e. $P(S_1)=P(S_2)=...=P(S_M)$. Therefore, we have to find any of these probabilities in order to find the others. Note that the number is equal to $L1_f$. Let P_{suc} be any of the successful transmission states. We then find P_{suc} by making the following steps. First, we find an expression for the normalization condition of the station model in a form that includes all state probabilities. We then solve the normalization condition to include the transition probabilities in the expression. By making simplifications, we write all the state probabilities in terms of P_{suc}. Thus, we end up with an expression with P_{suc} as a function of the transition probabilities.

Now let the transition probability b be the probability that the channel was busy in the last time slot and the transition probability c_f be the failure probability. Furthermore, let z_f be the probability to find the queue empty at the time of transmission. Finally, let g_f be the probability that at least one packet arrived from the upper layers in the last time slot.

Since the sum of the probabilities of being in all states is one, we have

$$\sum_{i=1}^{L1_f+x} P(S_i) + P(D) + \sum_{i=0}^{W-1} P(Y_i) + \sum_{i=0}^{M} P(B_{i0}) + \sum_{i=0}^{M}\sum_{j=1}^{V_i-1} P(B_{ij}) + \sum_{i=0}^{M}\sum_{j=1}^{L2_f} P(U_{ij}) + \sum_{i=0}^{M}\sum_{j=1}^{r} P(AT_{ij}) = 1, \tag{2}$$

where P(X) denotes the probability of being in state X of the station model.

We then find P_{suc} as a function of the transition probabilities b, f_f, z_f, g_f. Thus,

$$\begin{aligned}\frac{1}{P_{suc}} = L1_f + x + \frac{z_f b\left(1+\frac{W-1}{2(1-b)}\right)}{1-c_f^{M+1}} + \frac{z_f}{g_f\left(1-c_f^{M+1}\right)} \\ + \frac{4bc_f - 2(b+2c_f) - 4bc_f^{M+2} + 2bc_f^{M+1} + W\left(1-2^m c_f^{m+1}\right)}{2(1-b)\left(1-c_f^{M+1}\right)(1-c_f)(1-2c_f)} \\ + \frac{1+2c_f^{M+2} - c_f^{M+1} - W_f c_f + W\left(2^{m+1}c_f^{m+2} - 2^m + c_f^{m+1}\right)}{2(1-b)\left(1-c_f^{M+1}\right)(1-c_f)(1-2c_f)} \\ + \frac{c_f\left(L2_f + \frac{1-(1-b)^r}{b}\right)}{1-c_f}.\end{aligned} \tag{3}$$

Finally, we find the throughput from the following expression.

$$Throughput_f = P_{suc}.L1_f DataRate_f. \tag{4}$$

Note that expressions for the transition probabilities of the station model have not been found yet. We have to find their values to compute the throughput from equation 4. The transition probabilities of the station model are found in the following.

The probability of a busy channel

We first find the probability b that the channel was busy in transmitting successful or corrupted frames. We solve for b by using similar steps as we did before in finding P_{suc} but we use the normalization condition of the channel model instead.

Since the sum of the probabilities of being in all states is one, we have

$$P(E)+\sum_{i=1}^{L2_F+DIFS}P(N_{ffi})+\sum_{i=1}^{L2_S+DIFS}P(N_{SSi})+\sum_{i=1}^{L2_S+DIFS}P(N_{Sfi})+\sum_{i=1}^{L1_f}P(O_{fi})+\sum_{i=1}^{x}P(OK_{fi})+\sum_{i=1}^{L1_s}P(O_{si})+\sum_{i=1}^{x}P(OK_{si})=1. \quad (5)$$

where P(X) denotes the probability of being in state X of the channel model.

We then use equation 5 to find an expression for b as a function of the transition probabilities of the channel. Note that b is equal to 1 – (the probability that the channel was in the idle state E). Hence,

$$1/(1-b)=1+(L1_f+x+1)q_f+(L1_s+x+1)q_s+(L2_f+DIFS+1)p_{ff}+(L2_s+DIFS+1)p_{ss}+(L2_s+DIFS+1)p_{sf}. \quad (6)$$

Next, we find the probability τ_f that a station initiates a transmission in the next time slot given that the channel was free. From the station model, we find that it equals the probability that the backoff counter is decremented to zero and the station is in one of the first backoff states at any stage or it is in the idle state and received a new packet from the upper layers. Thus,

$$\tau_f=\frac{1}{1-b}\left[P(Y_0)+g_fP(D)+\sum_{i=1}^{M}P(B_{i0})+(1-z_f)P(B_{00})\right]=\frac{P_{suc}\left(\frac{z_fb}{1-c_f^{M+1}}+\frac{1}{1-c_f}\right)}{1-b}. \quad (7)$$

For a given number of fast stations n_f and slow station n_s, we find the transition probabilities of the channel model as follows. Let q_f be the probability that only one fast station initiates a transmission and no slow station initiates a transmission. Then,

$$q_f=n_f\tau_f(1-\tau_f)^{n_f-1}(1-\tau_s)^{n_s}. \quad (8)$$

Let q_s be the probability that only one slow station initiates a transmission and no fast station initiates a transmission. Then,

$$q_s=n_s\tau_s(1-\tau_s)^{n_s-1}(1-\tau_f)^{n_f}. \quad (9)$$

Let p_{ff} be the probability that two or more fast stations initiate transmissions and no slow station initiates a transmission. Then,

$$p_{ff} = \left(1 - n_f \tau_f \left(1 - \tau_f\right)^{n_f - 1} - \left(1 - \tau_f\right)^{n_f}\right)\left(1 - \tau_s\right)^{n_s}. \tag{10}$$

Let p_{ss} be the probability that two or more slow stations initiate transmissions and no fast station initiates a transmission. Then,

$$p_{ss} = \left(1 - n_s \tau_s \left(1 - \tau_s\right)^{n_s - 1} - \left(1 - \tau_s\right)^{n_s}\right)\left(1 - \tau_f\right)^{n_f}. \tag{11}$$

Finally, let p_{fs} be the probability that one or more fast stations initiate transmission and one or more slow stations initiate transmission. Then,

$$p_{sf} = \left(1 - \left(1 - \tau_f\right)^{n_f}\right)\left(1 - \left(1 - \tau_s\right)^{n_s}\right). \tag{12}$$

The Probability of Packet Loss

In a wireless network, the packet is lost if any of the following events occur. The packet is lost if it encounters a collision with another packet due to simultaneous transmission of two or more stations. It may also be lost if it is corrupted by errors during transmission on the channel due to fading, noise, interference, etc. Furthermore, a packet that encountered collisions after M transmission trials is dropped from the queue. Finally, a packet that joins a queue which is full is also dropped.

In our model, we assume that the queue at each station is large and there are no dropped packets due to a queue being full. We also explained earlier how to take into account the third possibility by including the number of transmission attempts M in the station model.

To compute the probability of a packet loss due to transmission failure because of collisions or channel errors, we define the failure probability c_f of a fast station as follows:

$$c_f = c_{col,f} + (1 - c_{col,f}) PER_f. \tag{13}$$

Similarly, for a slow station

$$c_s = c_{col,s} + (1 - c_{col,s}) PER_s. \tag{14}$$

The probability that a fast station encounters a collision is the probability that in any time slot, at least one of the remaining $(n_f - 1)$ fast stations or n_s slow stations transmits. Hence,

$$c_{col,f} = 1 - \left(1 - \tau_f\right)^{n_f - 1}\left(1 - \tau_s\right)^{n_s}. \tag{15}$$

Similarly, for a slow station

$$c_{col,s} = 1 - \left(1 - \tau_s\right)^{n_s - 1}\left(1 - \tau_f\right)^{n_f}. \tag{16}$$

Note that the calculation of PER, in a WLAN is hard to obtain analytically. As discussed in [7], approximations should be made to derive a close form of the PER as a function of the SNR. This motivated us to employ the study of the physical layer that was made in [3] for various types of coding and embed their results in our model.

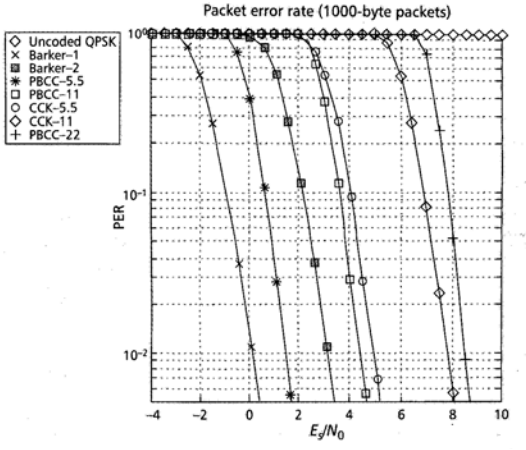


Fig. 3. The packet error rate (PER) as a function of the SNR for various types of coding and modulation schemes

Fig. 3 shows the PER of different types of coding and modulation schemes as a function of the received signal-to-noise ratio as it is illustrated in [3].

In Wireless LANs, The IEEE 802.11 standard uses Direct Sequence Spread Spectrum (DSSS) with a data rate 1Mbps when the modulation is the Binary Phase shift keying (BPSK) and 2 Mbps when it is the Quadratic phase shift keying (QPSK). It also uses a Baker code as the coding scheme. The 802.11b is an extension to the 802.11 standard using the same modulation types while providing higher data rates of 5.5 and 11Mbps using two different coding schemes. One code uses a short block length code, known as complementary code keying (CCK), and the other code incorporates a 64-state packet-based binary convolutional code (PBCC). The main difference between the two involves the much larger coding gain of the PBCC over CCK at a cost of additional computations at the receiver.

We found so far expressions for the transition probabilities b and c_f of the station model. We still need to find expressions for z_f and g_f as well.

For mathematical expediency, the arrival process has a Poisson distribution. The model can be extended assuming any other arrival process as the expressions of the channel and the station models can include any arrival process.

The system is analyzed using the M/G/1 queuing model. The consideration of other queuing models is possible but requires the distribution of the time ξ that a packet spends at the MAC layer before being correctly transmitted. This distribution is difficult to obtain without approximation. Using a M/G/1 queue, the average of this time is needed. This average can be found easily and accurately.

The properties of the M/G/1 queue affect the computation of the probability to find the transmission queue of a station empty. In an M/G/1 queue, the probability to find the queue of a station empty is

$$z_f = \max[0, 1 - \lambda_f . E_f(t)], \tag{17}$$

where λ_f is the frame arrival rate.

The probability that at least one frame arrived at the queue during the last time slot is 1- (the probability that no frame arrived). Hence,

$$g_f = 1 - e^{-\lambda_f}. \tag{18}$$

The average service time of a frame was derived in details in [2] and found as

$$\mathrm{E_f(t)} = (1-z)\mathrm{E_f(t_1)} + z\mathrm{E_f(t_2)}, \tag{19}$$

Where,

$$E_f(t_1) = (L1_f + x)(1 - c_f^{M+1}) + \left(c_f \frac{1 - c_f^{M+1}}{1 - c_f}\right)\left(L2_f + \frac{1-(1-b)^r}{b}\right) +$$

$$\left(\frac{1 - c_f^{M+1}}{1 - c_f}\right)\left(\frac{(1-2b)}{2(1-b)}\right) + \left(\frac{1-(2c_f)^{m+1}}{1-2c_f}\right)\left(\frac{W}{2(1-b)}\right) + \left(2^m c_f^{m+1}\right)\left(\frac{1 - c_f^{M-m}}{1 - c_f}\right) \tag{20}$$

and

$$\mathrm{E_f(t_2)} = \mathrm{E_f(t_1)} + (1-b)(c_f^{M+2} - 1)\left(1 + \frac{W-1}{2(1-b)}\right) \tag{21}$$

To find the values of the variables in the equations, we solve all the expressions numerically using the fsolve command in the optimization toolbox of MATLAB. Other numerical tools and methods can be used.

5 Numerical Results

We performed experiments and simulations to validate our MAC model in [2]. The numerical results we obtained from our model match what we got from the experiments. Since, the impact of the various modulation and coding schemes on the MAC throughput is represented by the addition of PER as a loss factor in the equation of the probability of packet loss and without changing the rest of the equations and parameters, the validation of the model is still applicable.

We now analyze the system by considering the channel aspects in evaluating its performance. Note from the graphs of Fig. 3 that the PER differs when the same type of coding and modulation is used while stations transmit at different data rates. It is inaccurate when studying the performance of the system to assume that the values of the PER are the same for fast and slow stations as was done in the literature. Our model differentiates between stations according to their data rates. Thus, we were able to assign appropriate values for each of them.

Fig. 4 shows the throughput experienced by stations for various types of coding and modulation schemes. Number of stations is set to five for illustrated purposes only. The horizontal axis represents SNR instead of PER to highlight the relation between the SNR at the physical layer and the throughput achieved at the MAC layer. Note that each SNR value is taken from its corresponding PER value of Fig. 3. As

seen in the graph, the PBCC outperforms CCK at low SNR values. The graph also shows the advantage of using low data rates at low SNR when stations need to communicate. Note that the stations would not be able to transmit any frame if a high data rate is chosen.

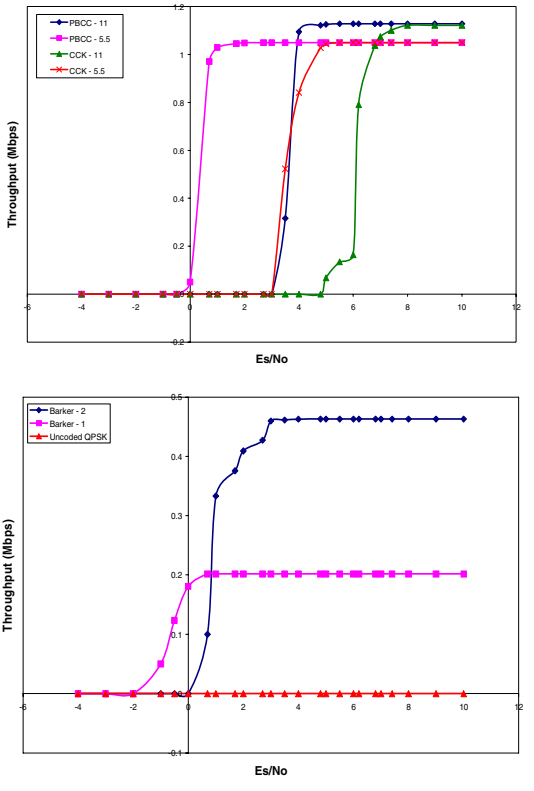


Fig. 4. Throughput per station at the MAC level as a function of SNR for various types of coding and modulation schemes and at different data rates

6 Conclusion

In this paper, we analyzed and studied the performance of IEEE 802.11 wireless LANs in a cross layer environment. We continued our previous work in [2] by including all the factors that affect the achieved MAC throughput in an expanded cross-layer-based model. Using the introduced model, we were able to obtain a direct relationship between the SNR at the physical layer and the throughput at the MAC sub-layer. We assigned appropriate values of SNR according to the transmission speed of the station since our model analyzes multi-rate wireless LANs. We studied the impact of various modulation and coding schemes on the MAC throughput and showed the advantage of using PBCC over CCK at low SNR values.

References

[1] IEEE Computer society. 802.11: Wireless LAN medium access control and physical layer specifications (June 1997)
[2] Abu-Sharkh, O., Tewfik, A.: Multi-rate 802.11 WLANs. In: IEEE Globecom, Saint Louis (November 2005)
[3] Heegard, C., et al.: High-Performance Wireless Ethernet. IEEE Communication Magazine (November 2001)
[4] Bianchi, B.: Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications 18(3) (2000)
[5] Chatzimisios, P., Vitsas, V., Boucouvalas, A.C.: Performance Analysis of IEEE 802.11 DCF in Presence of Transmission Errors. In: ICC 2004, Paris (2004)
[6] Helkov, Z., Spasenovski, B.: Saturation Throughput-Delay Analysis of IEEE 802.11 DCF in Fading Channel. In: ICC 2003 (2003)
[7] Manshaei, M., Cantieni, G., Barakat, C., Turletti, T.: Performance Analysis of the IEEE 802.11 MAC and Physical Layer Protocol. In: WoWMoM, Taormina (June 2005)
[8] Abu-Sharkh, O., Tewfik, A.: Cross-layer-based Modeling of IEEE 802.11 Wireless LANs with MIMO Links. In: IEEE Globecom, San Francisco (November 2006)

Improving the Efficiency of the Proxel Method by Using Individual Time Steps

Claudia Krull[1], Robert Buchholz[2], and Graham Horton[1]

[1] Intitut für Simulation und Graphik, Otto-von-Guericke-University, Universitätsplatz 2, Magdeburg 39106, Germany
[2] Institut für Technische und Betriebliche Informationssysteme, Otto-von-Guericke-University, Universitätsplatz 2, Magdeburg 39106, Germany

Abstract. Discrete stochastic models (DSM) are widely used in various application fields today. Proxel-based simulation can outperform discrete event-based approaches in the analysis of small stiff DSM, which can occur for example in reliability modeling. However, when parallel processes with largely differing speed are involved, the faster process determines the small discretization time step, investing far too much effort into the approximation of the slower process. This paper relieves that problem by using individual time steps for each transition and situation. The key problem is to keep semantic consistency when using different time steps for parallel transitions. However, the preservation of the probability mass in every single simulation time step could be achieved. Experiments show that binary step division in conjunction with appropriate subdivision criteria can outperform the original Proxel method significantly. This increases the applicability of Proxels, by enabling the analysis of larger and therefore more realistic models.

Keywords: Proxel-based simulation, discrete-time Markov chains, state space-based simulation, discrete stochastic models.

1 Introduction

Discrete stochastic models (DSM) are used in various application fields such as reliability modeling, manufacturing, planning and control. The analysis of DSM is usually done using discrete event-based simulation. This can become expensive when small stiff models are involved, such as in reliability modeling. Many replications might be necessary to obtain statistically significant results. Proxel-based simulation [1,2] can outperform traditional DES on small stiff models. It deterministically discovers all possible system developments and their probabilities in discrete time steps, avoiding a possibly large number of replications.

However, when transitions with largely differing speed are involved, the original Proxel algorithm cannot perform optimally. A globally fixed constant time step is either too small for a slow transition, investing more computational effort than necessary, or it is too large for a fast transition, producing a too large error. A small constant time step also leads to a more pronounced state space explosion, severely limiting the size of model that can be simulated effectively.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 116–130, 2009.

An ideal solution would be to use a different time step for each transition, adjusted to the speed of the transition, or even to the current situation. The paper shows an approach realizing the concept of individual time steps for each transition, these are also called variable time steps. It uses binary step division, providing maximum flexibility and computational efficiency. The key problem to tackle is a semantic inconsistency that arises when using different time steps for parallel transitions. By redistributing the probability completely in every step, the original Proxel-method ensures that the sum of the probability of all possible states at one time is still 1. This is no longer valid when using differently sized time steps in parallel. The solution proposed here computes all probabilities for the smallest time step necessary. For slower transitions this probability is collected in a container, which is processed further along the time scale.

Experimental results indicate that using an appropriate subdivision criterion, variable time steps can increase the accuracy of a Proxel simulation by several orders of magnitude, or accordingly decrease the computation time necessary to obtain the same accuracy. Extrapolation of the results using different thresholds for the subdivision criteria is only of limited applicability. The paper shows, that variable time steps can reduce state space explosion, eventually enabling the simulation of larger models and thereby increasing the applicability of Proxels. The theory and results presented here are the result of a Masters thesis [3].

2 Proxel-Based Simulation

The Proxel-based simulation algorithm was developed by Horton [1] and further improved by Lazarova-Molnar [2]. It is a state space-based simulation method, which is based on the method of supplementary variables [4]. This section gives a brief overview of the basic ideas involved in Proxel simulation.

A so-called Proxel (see Equation (1)) represents a probability element in the expanded state space of the model. It contains the discrete model state dS and the age vector τ of the active or race age transitions as coordinates in the expanded state space. It also includes the current point in simulation time t, the route to this particular Proxel R and the probability of that combination p. Route R and simulation time t are rarely explicitly included in practical implementations, t is usually global and by omitting R, the reachable state space is reduced considerably.

$$P = (S, R, p) = ((dS, \tau, t), R, p) \tag{1}$$

By extending the discrete system states with the discretized age of all currently enabled or otherwise interesting transitions, a non-Markovian process is turned into a discrete-time Markov chain (DTMC) [5]. The one-step transition probability between these DTMC states can be determined dynamically using the so-called instantaneous rate function (IRF), saving the effort of explicitly building the Markov chain.

The IRF as defined in Equation (2) represents the current rate of probability flow from one state to the next [1].

$$\mu(\tau) = \frac{f(\tau)}{1 - F(\tau)} \tag{2}$$

The main idea of the Proxel algorithm is a simple iterative approach. The initial Proxel $((S_0, \boldsymbol{\tau_0}, 1))$ contains the initial system state S_0, an all zero age vector τ_0 and the probability 1. All possible successive system states are determined and using the given discrete time step Δ, the corresponding transitions IRF μ and the transitions age in τ their probability at time Δ can be approximated, assuming at most one state change can happen within one time step. This procedure is then repeated for all follow-up Proxels generated at time Δ. Thereby a so-called Proxel tree is generated, which contains all possible system developments at discrete intervals. The number of Proxels at one point in time can become quite large, due to the possible number of combinations of values in the age vector, expanding the system state. This increases memory requirement as well as computation cost.

The basic assumption of the Proxel method is that at most one state change can happen within one time step, therefore the simulation result can only ever be an approximation. The time step plays a central role in the performance and accuracy of the method. The smaller the time step, the more accurate the result is, but also the more pronounced the state space explosion and the more expensive the computation. However, larger time steps can be used to extrapolate more accurate results using Richardson extrapolation [6].

The key problem of Proxels is the state space explosion. Thus, a time step as large as possible is desirable. However, if transitions of differing speed are involved, the faster transition determines the maximum possible step size, resulting in a time step that is smaller than necessary for the slower transition. The example in Figure 1 illustrates that behavior. It shows a small stochastic Petri net with two consecutive transitions that differ in speed by a factor of 10. Both have a normal distribution and the same coefficient of variation. Using the appropriate time step for the faster transition computes the result for the slower transition to an accuracy that is ten times as large. Using the appropriate time step for the slower transition reduces the accuracy of the faster transition by a factor of 10. Either too much is invested, or the error induced is too large. In this example, the ideal time steps of the two transitions would also vary by a factor of 10, approximating both with comparable accuracy and effort.

The constant global time step is one of the key performance factors in Proxel simulation, it influences accuracy, as well as cost. However, the accuracy achieved

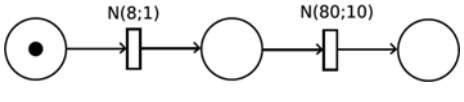


Fig. 1. Simple Example Model with Transitions of Differing Speed

using the same time step can be very different for transitions with largely differing speed. This paper tries to remedy this problem by using a separate time step for each transition, which is adapted to the individual transitions speed.

2.1 Error Sources in Proxel Simulation

Since the goal of individual time steps is to simulate transitions of different speed with comparable accuracy, one needs to be able to estimate the accuracy. Therefore the error sources of Proxels need to be identified and investigated.

BA Error. The first source of error in Proxel-based simulation is an error made knowingly through the basic assumption (BA). This assumption states that no more than one state change can be made within one discrete time step, neglecting the possibility of having two or more consecutive state changes within one step. Since this is one basic property of the simulation algorithm, the effect of the error can only be reduced by reducing the step size, but not by changing any algorithm feature. However, the error can be estimated by calculating the probability of two consecutive state changes in one step (see [3] for further details).

ODE Error. The second known source of error in Proxel-based simulation is the integration method used when estimating the one-step transition probabilities (ODE). The dynamic behavior of a single transition is defined by Equation (3), where $\Pi(t)$ represents the probability of the state at time t and $\mu(t)$ is the value of the instantaneous rate function (IRF) at time t. This means that the rate of change of the probability to stay in a state is proportional to the hazard rate function value and the remaining probability mass at that time. This is an ordinary differential equation and due to discretization the problem stated in (4) needs to be solved. Former Proxel implementations used Eulers method (5) or trapezoid integration (6) (Heuns method), which is not very accurate, but was sufficient for small time steps. When using larger time steps for slower functions, the error induced by an inappropriate integration method can no longer be disregarded. The error can however easily be reduced by using higher order integration methods such as Runge-Kutta. Using embedded Runge-Kutta methods, the error can even be estimated at little extra cost. One example of such a method is the combination of Euler's method and Heun's method, forming the ODE12 integration scheme. A more accurate, but also more expensive method was introduced by Dormand and Prince using fourth and fifth order integration methods, which is simply called ODE45. A more detailed description of the integration methods used in this paper can be found in [3].

$$\frac{d\Pi}{dt} = \Pi'(t) = -\Pi(t) * \mu(t) \tag{3}$$

$$\Pi(t + \Delta) = \Pi(t) + k * \Delta \tag{4}$$

$$k = -\Pi(t) \times \mu(t) \tag{5}$$

$$k = -\Pi(t) \times \frac{1}{2}(\mu(t) + \mu(t + \Delta) \tag{6}$$

ND Error. A third source of error has been examined in detail for the first time in [3]. It can be called error through the non-deterministic behavior of the method (ND). Proxels approximate the continuous flow of probability between two states by a step function, causing an unnatural behavior, which can have unexpected side effects. One of these effects is, that there is one time step delay in activation of the transitions leaving a subsequent state, even though when in real continuous time they would have been activated almost as soon as the simulation started. This error can be estimated by a more detailed computation of the transitional behavior within the time step. Experiments showed that it can be reduced by simply initializing the age of a new state to $\frac{1}{2}\Delta$ instead of 0 as was done before. This takes into account that the subsequent transition could in reality have been activated anytime within the time step and not at its end. This is still just an approximation, but showed to be efficient and effective.

One primary goal of variable time steps is to control and reduce these errors. Only then can larger time steps be used without unwanted loss of accuracy.

3 Introducing Variable Time Steps

This section first defines requirements of a Proxel algorithm using individual (variable) time steps. Then some basic properties of the algorithm and the new algorithm itself is described. The final part of the section details on the time step subdivision criteria as a central item of the VTS algorithm.

3.1 Requirements of a VTS Algorithm

A suitable algorithm for variable time steps (VTS) should ideally fulfill the following requirements:

- It can choose the ideal step size for each transition and current setting. Only then can the full potential of VTS be exploited.
- It can dynamically determine the step size during runtime using model and transition properties. This is more flexible than a pre-computation step, since it can also react to changing circumstances during runtime.
- It does not have too much overhead over constant time steps (CTS), regarding computation time and memory requirement. Too much overhead would reduce the advantage of VTS compared to CTS.
- It is still able to conserve probability by completely distributing it to all subsequent Proxels. It is necessary to locate the total probability mass in each time step of the simulation to compute statistics and result measures.

Some of these requirements are contradicting, such as maximum flexibility and minimum overhead, therefore we are looking for a good trade-off between them.

3.2 Properties of the VTS Algorithm

This section describes two key decisions made regarding the current implementation, which also distinguish it from a former attempt to use variable time steps in Proxel simulation [7].

The first decision was taken regarding the method of time step division. A choice providing maximum flexibility would be an arbitrary time division. This is however not suitable, since merging Proxels with the same age at the same point in simulation time is one central way to counter state space explosion. This would however no longer be possible using arbitrary step division. The compromise between computational efficiency and flexibility taken in this paper is to choose a binary step division scheme. This means for each transition leaving a Proxel to decide whether the current time step is small enough, according to some criterion, or whether it should be bisected and simulated in two steps. This ensures that Proxels can be merged at higher level division points, maximizing the number of joining points. Another effect is that it reduces the question of the ideal time step to a simple decision problem, which can be applied recursively. Therefore the decision can be made during runtime based on current model and transition properties.

The second decision was to chose the time step separately for each transition leaving a Proxel. This allows for maximum flexibility at additional computational effort. However, choosing different time steps for parallel transitions can result in semantic problems. It is no longer intuitively clear, how the probability should be distributed to the different child Proxels, which can now exist at different points in time. The conservation of the probability mass needs to be ensured at each simulation time step, enabling the consistent computation of transient statistics at the smallest steps taken. The solution chosen in this paper is to calculate the probability leaving the parent Proxel for the smallest time step chosen. For the transitions using longer time steps it is stored in containers for later handling.

3.3 New Proxel Simulation Using Variable Time Steps

This section briefly describes the current algorithm for Proxel-based simulation using individual time steps.

The simulation starts with an initial Proxel with probability 1, all transition ages set to 0 and in the initial discrete system state, as in the original Proxel algorithm [2]. Then the algorithm computes all possible follow-up Proxels, resulting from the firing of the transitions active in this state. In contrast to the original algorithm the time step, and therefore the point in future simulation time where the Proxel is created, is determined individually for each transition. Since binary step division is used, the algorithm starts with a single time step reaching to the end of simulation time and bisects it recursively. The time step is then fixed for each transition out of the current Proxel and repeated until no more probability is left in the original Proxel's marking. For each Proxel generated this scheme is repeated, except that the maximum time step for a transition is the next larger step taken on the higher levels. This ensures the possibility of merging Proxels later on in the simulation time.

The algorithmic handling of different parallel time steps is pictured in Figure 2 (right). The SPN of the model being simulated is shown in Figure 2 (left). The model has three parallel transitions, the transition leading to place B is the fastest of the three, the transition to place C is two times slower and the

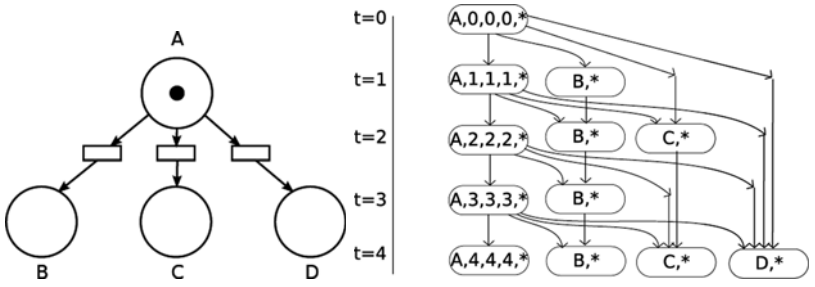


Fig. 2. Example SPN of Model with Three Parallel Transitions of Differing Speed (left) and Computing Successor Proxels for Example Model (right)

transition leading to state D four times slower than AB. Figure 2 (right) shows the start of the Proxel tree resulting from the simulation of this SPN. Each Proxel contains the following information: the current discrete system state, the age of the three activated transitions in state A and an asterisk as a placeholder for the probability of that expanded system state.

The discrete time steps chosen for the three transitions to places B, C, and D are 1, 2 and 4. The Proxels for staying in place A and entering place B are created in every time step. The probability to leave state A for the states C and D is also computed for each step of size 1, however the Proxels are only created at time 2 and 4 respectively. The probability leaving A in one time step is stored in these containers until the appropriate time step is reached.

In this way one can ensure that the sum of the probability of all current and future Proxels created out of a single Proxel corresponds to the Proxels original probability. In each time step the location of the total probability mass is clearly defined. The additional computational effort to compute all outgoing probability at the minimal step needs to be invested to ensure semantic unambiguousness and conservation of the probability mass. To reduce the effort of determining the memory location and retrieval of future Proxels, the definition of a Proxel is extended by a list of redirectors and redirector recursion depth. With the help of these, the target Proxel for each transitions probability can be determine using constant effort. The global storage of the Proxels in different time steps is realized by a hierarchy of Proxel containers, one for each level of the recursion. Experiments showed that no more than 20 recursion levels were required to simulate the tested models to a sufficient accuracy.

3.4 Time Step Subdivision Criteria

One crucial point of a good algorithm for variable time steps using binary step division is a good subdivision criterion. Five different criteria were developed in [3], which will be described here and tested in the experiments section 4. The main goal is to produce comparable accuracy for the simulation of transitions with differing speed. Therefore, the subdivision criteria try to achieve a similar

error in every simulation time step. All of the criteria use certain thresholds to determine whether the time step should be subdivided or not.

GLOBAL_PROB. A naive approach of a flexible time step size would be to set a global threshold for the maximum probability of a Proxel (GLOBAL_PROB). This would mean to subdivide a time step until the resulting child Proxels have a probability below the given threshold. This seems to be a reasonable criterion, since the smaller the overall probability of a Proxel the smaller the possible contribution to the global simulation error. This however fails when processing the follow-up Proxels created for the initial Proxel. All of these have a probability below the global threshold, therefore their outgoing transitions can be simulated using the maximum possible time step, since the probability of the subsequent Proxels can never exceed that of the parent Proxel. This is not the desired behavior, and therefore (GLOBAL_PROB) is not applicable.

TRANS_PROB. Another simple idea is to limit the probability of a one-step state transition (TRANS_PROB). This threshold limits the fraction of probability that can leave a Proxel (discrete system state) within one time step. This also makes sense, since using smaller time steps also reduces the possible error made in one step. However, as the probability of the original Proxel is reduced, the fraction reduces accordingly. If in each step half of the probability is allowed to leave the Proxel, it will theoretically never loose all of its probability and the time steps will become smaller, eventually preventing an advancing of simulation time beyond the support of the distribution. The Proxel algorithm however prevents this loop, since it discards Proxels below a given probability threshold. Therefore, (TRANS_PROB) will be investigated further.

ORIG_PROB. The third criterion limits the amount of probability leaving a Proxel within one step to a given fraction of the original Proxel probability (ORIG_PROB). This ensures that the time step is not decreased indefinitely as can happen for TRANS_PROB and that all of the probability leaves the Proxel eventually. It also ensures similar accuracy in all successor Proxels and therefore is a promising candidate for a subdivision criterion.

MEAN. The fourth criterion is based on a finding documented in [2], where it was stated that the maximum time step allowed should not exceed one half of the fastest transitions mean value. Generalizing this we define the MEAN criterion: the time step size is not allowed to exceed the fraction k of the transitions distribution mean. This rather simple criterion ensures about equal accuracy of all distribution approximation, scales with the transitions speed and prevents an indefinite subdivision of the time step.

UNANIMOUS. The fifth criterion developed in [3] is based on the three error sources of Proxel-based simulation (see Section 2.1), which were described in Section 2.1. The criterion UNANIMOUS is used in such a way, that it no longer subdivides a time step when all three error methods do not exceed a given

threshold anymore. This should directly limit the error made in each time step to the given threshold.

All criteria except the GLOBAL_PROB seem to be suitable candidates for a valid subdivision criterion. Therefore they need to be tested for their applicability and performance regarding runtime and accuracy. All of these criteria are based on threshold values. Therefore, an extrapolation to a small threshold value using rougher estimates computed using larger thresholds should be theoretically possible. A formal proof of linear convergence of the results when reducing the threshold k in a given criterion was not possible. Therefore experiments were conducted to test the applicability of the Richardson extrapolation.

4 Experimental Results

This section describes some of the experiments conducted using the newly developed Proxel-based algorithm using variable time steps. The experiments are selected from [3], for more detailed results and further experiments the reader is referred there.

The first set of experiments (see Section 4.1) tests several combinations of subdivision criteria and integration methods. We expect that at least some of the criteria can reliably outperform constant time steps by attaining better accuracy with comparable computational effort. The second set of experiments (see Section 4.2) tests the applicability of Richardson extrapolation using the thresholds of the subdivision criteria.

The general experiment setting used was the following:

- Trapezoid integration for the approximation of one-step transition probability. (see Equation (6))
- As an exception, for the UNANIMOUS criterion embedded Runge-Kutta methods (ODE12, ODE45) were necessary to estimate the integration error per step.
- The age of a newly created Proxel was initialized with $\frac{1}{2}\Delta$, to reduce the ND-error (see Section 2.1).
- Proxel cutoff probability was set to 10^{-15}, meaning that Proxels with a probability below that threshold were discarded, in order to dampen state space explosion.

In earlier papers the comparison criterion between different approaches was always the effort invested and accuracy obtained using a given time step size. Since we are now looking at dynamically determined time step sizes, the performance criterion needs to be generalized. The comparison between different algorithm configurations happens according to the accuracy that could be obtained over the computational effort invested. The accuracy is determined by the error of the Proxel result compared to the analytically obtained solution. The models chosen for testing were all simple enough to obtain an analytical solution for the results of interest. Neither the solution accuracy nor the computational effort can be controlled directly, both are a result of algorithm runtime behavior.

Thus, the algorithms can not be compared at specific points and the individual measurements need to be interpolated to yield comparable plots.

4.1 Which Subdivision Criterion to Use?

The first experiment was conducted using a simple chain-like model (see Figure 3) with five successive transitions of decreasing mean value. The distributions of the successive state transitions are the following: $N(1;0,25)$, $N(4;0,5)$, $N(9;0,75)$, $N(16;1)$ and $N(25;1,25)$. The result measure of interest in the chain model was the average time of the last transition firing.

The accuracy over runtime results for the criteria ORIG_PROB, MEAN and TRANS_PROB as well as for constant time steps (CTS) are depicted in Figure 4. The results of all other criteria tested were omitted from the diagram, because they did not converge to the actual result value (which was determined analytically). In this diagram the lowest curve shows the most efficient algorithm, since here the least effort was needed to achieve a certain level of accuracy.

The constant time steps exhibit a reliable behavior, the more computational effort is invested, the smaller the error of the result. Only the ORIG_PROB criterion can outperform the constant time steps, and it does so by about two orders of magnitude. Hence, the plot looks as if it were a straight line. The MEAN criterion can almost compete with the performance of CTS, but has a slightly higher effort to obtain comparable accuracy. The TRANS_PROB criterion behaved reliably, increasing the accuracy with increased effort, but could not reach the efficiency of constant time steps.

The second experiment was conducted using a warranty model based on a real life Proxel application also discussed in [2]. The reachability graph of the model is shown in Figure 5. The goal is to compute the warranty cost for a given configuration of year and mileage based warranty expiration, a given failure distribution function and cost per failure. The measure of interest in the warranty model was the average cost incurred within the warranty period.

The result graph includes results for constant time steps and the following subdivision criteria: ORIG_PROB, MEAN, TRANS_PROB, UNANIMOUS using ODE12 embedded integration method (Unanimous-ODE12) and ODE45 embedded integration method (Unanimous-ODE45). In this experiment again, CTS showed reliable behavior, even though not as linearly-looking as for the chain model. The constant time steps were only outperformed by the MEAN criterion and by ORIG_PROB, with the latter one being the most efficient criterion overall. The two settings of the UNANIMOUS criterion could not outperform CTS,

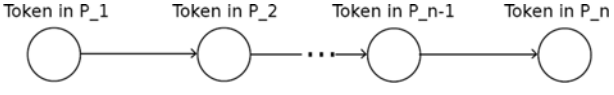


Fig. 3. Reachability Graph of Chain Model

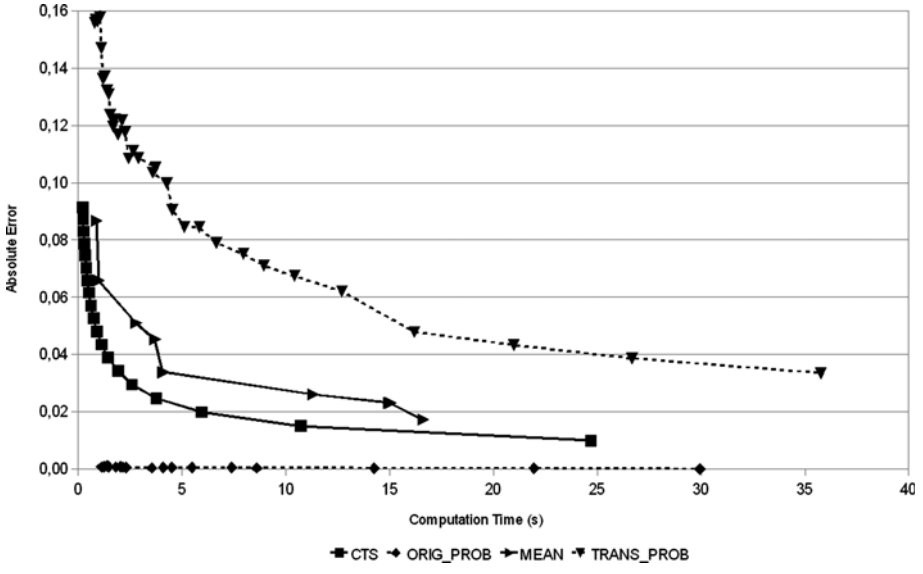


Fig. 4. Accuracy Obtained Over Computation Time for Chain Model and Different Subdivision Criteria

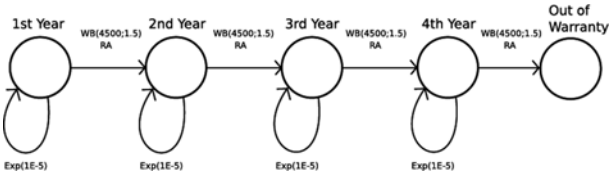


Fig. 5. Reachability Graph of Warranty Model

even though the combination with ODE45 is comparable to constant time steps regarding the results.

Result discussion. The experiments comparing the different time step subdivision schemes showed that few of them could reliably outperform constant time steps. For further experiments refer to [3]. Only the ORIG_PROB criterion was better than CTS in all experiments. MEAN was competitive, but not always better than CTS. The TRANS_PROB criterion performed worse than CTS on some models. The UNANIMOUS criterion did not converge to the analytical result for any model and can not be used reliably. This poor performance is probably due to a combination of still poor understanding of the exact effects of the different errors and resulting improper estimation. Furthermore it is not clear how to combine three totally different error effects using only one threshold, or what the combination of three different thresholds could be. A competitive UNANIMOUS criterion requires further research effort on these topics. Overall the results show

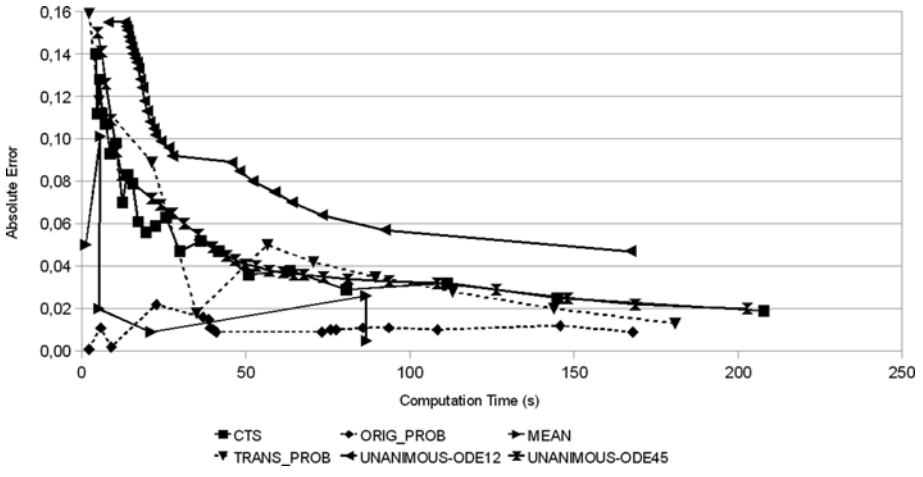


Fig. 6. Accuracy Obtained over Computation Time for Warranty Model and Different Subdivision Criteria

that individual (variable) time steps can outperform constant time steps, when an appropriate time division criterion is chosen (here ORIG_PROB).

4.2 Applicability of Richardson Extrapolation

This section presents some experimental results in the applicability of Richardson extrapolation. Only MEAN and ORIG_PROB were tested, due to their constant and reliable performance in earlier experiments. The extrapolation was done using the threshold value k for the fraction of the transitions distribution mean (MEAN criterion) and the fraction of the original Proxel probability (ORIG_PROB criterion). Both thresholds were varied between 1 and 0, where 1 resulted in too rough results and 0 was excluded for obvious reasons. Again these experiments are only a selection of the results shown in [3]. The graphs show the actual result measures for each of the models for threshold values between 0 and 1, and for better resolution of small values also between 0 and 0.3.

The convergence behavior was tested on the chain and warranty models already described in the previous section. Using the MEAN subdivision criterion, the results converge to the analytical solution, however not linearly. The solutions of the warranty model alternate around the real value in successively smaller jumps (see Figure 8) and the solution of the chain model converges as a step function (see Figure 7). This is not a problem of the criterion, but a side effect of the choice of binary subdivision of the steps. The model transitions change step size by an order of 2 only at certain values of k and stay stable in between. This behavior is not perfect, but using an intelligent choice of threshold pairs (k and $\frac{1}{2}k$) extrapolation should be possible using the MEAN criterion.

The convergence behavior of the ORIG_PROB criterion is less reliable. Even though the results do converge toward the analytical solution, this happens in

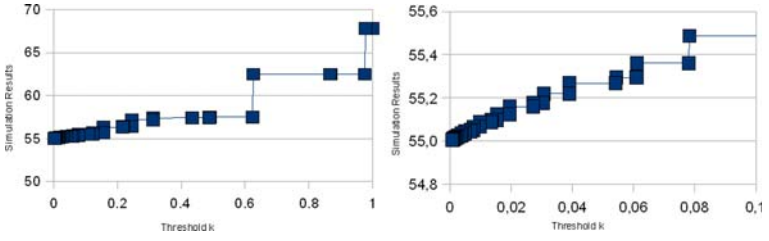


Fig. 7. Result Extrapolation for Chain Model and MEAN Criterion

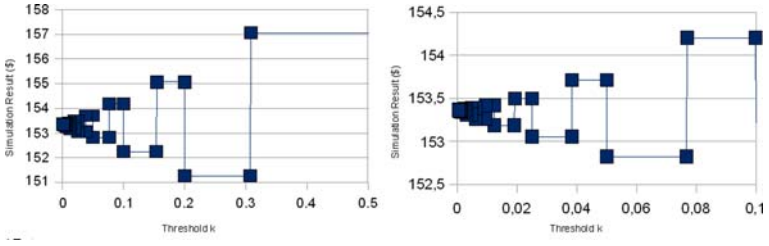


Fig. 8. Result Extrapolation for Warranty Cost and MEAN Criterion

an erratic fashion. The result of the chain model converges in seemingly unpredictable jumps (Figure 9), making extrapolation more a lottery game than reliable. The result of the warranty model seems to converge almost linearly for larger threshold values (Figure 10), the magnification of the smaller values however also shows unpredictable behavior. Therefore, extrapolation using the ORIG_PROB criterion is not reliable, an arbitrary combination of points might point to the wrong final result.

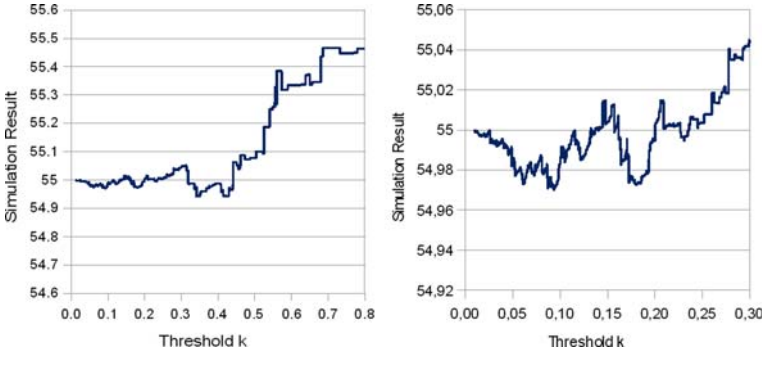


Fig. 9. Result Extrapolation for Chain Model and ORIG_PROB Criterion

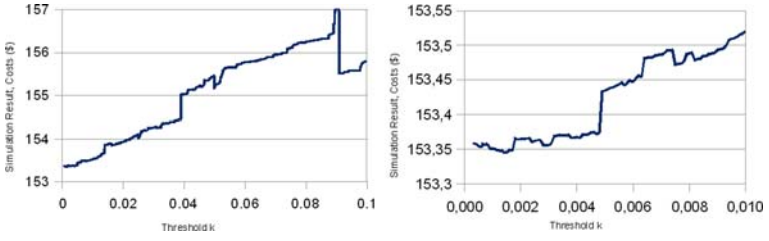


Fig. 10. Result Extrapolation for Warranty Cost and ORIG_PROB Criterion

Result discussion. The experiments on extrapolation using the threshold values of the subdivision criteria were only partially successful. The simulation results did not converge linearly for either one of the two criteria. Using the MEAN criterion, a good choice of the threshold values to extrapolate is a pair of values k and $\frac{1}{2}k$. Only then can one be sure that all model transitions are using twice the step size for k than for $\frac{1}{2}k$. The results obtained using the ORIG_PROB subdivision criterion converged erratically, and do not seem to exhibit any structure. One problem could be that the binary step division forces the transition step size to change in jumps instead of a smooth transition, however that is a key point of the VTS algorithm presented and should not be changed.

5 Conclusion and Outlook

The paper showed an approach to tackle the problem of state space explosion in Proxel-based simulation. Instead of the original constant time steps, time step size was chosen dynamically for each single transition at runtime, enabling an optimal step size in each situation. The choice for binary subdivision resulted in an efficient algorithm with little computational overhead compared to constant time steps. However, the choice of subdivision criterion is crucial to the performance. Only two of the tested criteria could reliably outperform constant time steps on the tested stiff models. The better of the two criteria could achieve the same accuracy about 100 times faster than the constant time step algorithm. Therefore the goal of the paper has been reached. Based on the experiments and further experiments in [3] the extension of individual or variable time steps increases the competitiveness of Proxels in real world applications, making the algorithm feasible for larger and more realistic models.

Future work. More work is required in the future to enhance the competitiveness of variable time steps in Proxels. A better error estimation could lead to more efficient subdivision criteria, as could a structural analysis of the model to be simulated. To dampen state space explosion further when the time steps get very small, one could start merging Proxels with similar, not just exactly matching age vectors. The choice of integration method can also affect the algorithm performance and should be observed more closely in the future.

References

1. Horton, G.: A new paradigm for the numerical simulation of stochastic petri nets with general firing times. In: Proceedings of the European Simulation Symposium 2002, pp. 129–136. SCS European Publishing House (2002)
2. Lazarova-Molnar, S.: The Proxel-Based Method: Formalisation, Analysis and Applications. PhD thesis, Otto-von-Guericke-University Magdeburg (2005)
3. Buchholz, R.: Improving the Efficiency of the Proxel Method by using Variable Time Steps. Master's thesis, Otto-von-Guericke-University Magdeburg (2008)
4. German, R., Lindemann, C.: Analysis of stochastic petri nets by the method of supplementary variables. In: Proceedings of Performance Evaluation, vol. 20, pp. 317–335 (1994)
5. Bolch, G., Greiner, S., de Meer, H., Trivedi, K.S.: Queuing Networks and Markov Chains. John Wiley & Sons, New York (1998)
6. Richardson, L.: The deferred approach to the limit. part i. single lattice. Philosophical Transactions of the Royal Society of London, Series A 226, 817–823 (1927)
7. Wickborn, F., Horton, G.: Feasible state space simulation: Variable time steps for the proxel method. In: Proceedings of the 2nd Balkan Conference in Informatics, Ohrid, Macedonia, pp. 446–453 (2005)

Efficient On-Line Generation of the Correlation Structure of F-ARIMA Processes

Maria-Estrella Sousa-Vieira, Andrés Suárez-González,
José-Carlos López-Ardao, and Cándido López-García

Department of Telematics Engineering, University of Vigo, Spain
estela@det.uvigo.es

Abstract. Several traffic measurement studies have shown the presence of persistent correlations in modern networks. The use of stochastic processes able to capture this kind of correlations, as self-similar processes, has opened new research fields in network performance analysis, mainly in simulation studies, where the efficient synthetic generation of samples is one of the main topics. Although F-ARIMA processes are very flexible to capture both short- and long-range correlations in a parsimonious way, only off-line methods for synthesizing traces are efficient enough to be of practical use. In order to overcome this disadvantage, in this paper we propose a M/G/∞-based efficient and on-line generator of the correlation structure of F-ARIMA processes.

Keywords: F-ARIMA processes, M/G/∞ process, Correlation, Synthetic efficient on-line generation.

1 Introduction

Several traffic measurement results have convincingly shown the existence of persistent correlations in the traffic of modern networks [12,21,3,28,5,8,19]. These experimental findings stimulated the opening of a new branch in the stochastic modeling of traffic, since the impact of the correlation on the performance metrics may be drastic [22,26,23,11]. The use of classes of stochastic processes for traffic modeling purposes, displaying forms of correlation as diverse as possible by making use of few parameters, as self-similar processes, is important. Usually, real traces are of limited length and lack the necessary diversity required to perform simulation studies.

Some of these processes are Fractional Gaussian Noise [25] (FGN), Fractional AutoRegressive Integrated Moving Average [14,16] (F-ARIMA) and M/G/∞ [6].

Unlike the FGN process, whose correlation structure, determined by a single parameter, is too rigid, F-ARIMA processes are very flexible to capture both short- and long-range correlations using few parameters. In fact, these processes have been widely used for modeling traffic sources [12,4,1]. Nevertheless, only off-line methods for synthesizing F-ARIMA traces are efficient enough to be of practical use.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 131–143, 2009.

The M/G/∞ process is a stationary version of the occupancy process of an M/G/∞ queueing model. In addition to its theoretical simplicity, it can be used to model different traffic sources, because it is flexible enough to exhibit both Short-Range Dependence (SRD) and Long-Range Dependence (LRD). Moreover, queueing analytical studies are sometimes feasible [10,35,30,27], but when they are not, it has important advantages in simulation studies [20,29], such as the possibility of on-line generation and the lower computational cost (exact methods for the generation of a trace of length n require only $O(n)$ computations).

In this paper, we propose an efficient and on-line generator of M/G/∞ processes for matching the correlation structure of F-ARIMA processes.

The remainder of the paper is organized as follows. In Section 2 we review the main concepts related to SRD, LRD and self-similarity. In Section 3 we describe F-ARIMA processes and the most important methods for synthesizing traces that have been proposed. The M/G/∞ process is described in Section 4. We also remind briefly the method that we have presented in [31,32] to improve the efficiency of the generator when the distribution of the service time of the M/G/∞ system has subexponential decay. In Section 5 we explain the main concepts related to the Whittle estimator that we are going to use in order to compare FGN and F-ARIMA processes for VBR video traffic modeling purposes. In Section 6 we present the M/G/∞-based generator of the correlation structure of F-ARIMA processes, and we evaluate its efficiency and the quality of the samples, and in Section 7 we apply it to the modeling of the correlation structure of VBR video traffic, and we use a method based on the prediction error of the Whittle estimator to choose the best among several processes. Finally, in Section 8 we summarize the conclusions.

2 SRD, LRD and Self-similarity

It is said that a process exhibits SRD when its autocorrelation function is summable, i.e., $\sum_{k=1}^{\infty} \mathrm{r}_k < \infty$, like in those processes whose autocorrelation function decays exponentially:

$$\exists \alpha \in (0,1) \left| \lim_{k\to\infty} \frac{\mathrm{r}_k}{\alpha^k} = \mathsf{c}_\mathrm{r} \in (0,\infty) \right. .$$

Its spectral density is bounded at the origin.

Conversely, it is said that a process exhibits LRD [7] when its autocorrelation function is not summable, i.e., $\sum_{k=1}^{\infty} \mathrm{r}_k = \infty$, like in those processes whose autocorrelation function decays hyperbolically:

$$\exists \beta \in (0,1) \left| \lim_{k\to\infty} \frac{\mathrm{r}_k}{k^{-\beta}} = \mathsf{c}_\mathrm{r} \in (0,\infty) \right. . \tag{1}$$

Its spectral density has a singularity at the origin.

Let $X = \{X_n; n = 1, 2, \dots\}$ be a stationary stochastic process with finite variance and let $X^{(m)}$ be the corresponding aggregated process, with aggregation

level m, obtained by averaging the original sequence X over non-overlapping blocks of size m: $X^{(m)} = \left\{\overline{X_i}[m]; i = 1, 2, \dots\right\}$, where:

$$\overline{X_i}[m] = \frac{1}{m} \sum_{n=(i-1)m+1}^{im} X_n.$$

The covariance stationary process X is called exactly second-order self-similar, with self-similarity parameter H [18], if the aggregated process $X^{(m)}$ scaled by $m^{1-\mathsf{H}}$ has the same variance and autocorrelation as X for all m, that is, if the aggregated processes has the same non-degenerate correlation structure as the original stochastic process.

The autocorrelation function of both X and $X^{(m)}$ is:

$$\mathrm{r}_k = \mathrm{r}_k^{\mathsf{H}} \stackrel{\Delta}{=} \frac{1}{2}\left[(k+1)^{2\mathsf{H}} - 2k^{2\mathsf{H}} + (k-1)^{2\mathsf{H}}\right] \quad \forall k \geq 1, \tag{2}$$

where: $\lim_{k\to\infty} \frac{\mathrm{r}_k^{\mathsf{H}}}{k^{2\mathsf{H}-2}} = \mathsf{H}(2\mathsf{H}-1)$, that is, it decays hyperbolically as in (1), with $\beta = 2 - 2\mathsf{H}$, and so the process exhibits LRD if $\mathsf{H} \in (0.5, 1)$.

If (2) is satisfied asymptotically by the autocorrelation function of the aggregated process, $\mathrm{r}_k^{(m)}$, then the process is called asymptotically second-order self-similar:

$$\lim_{m\to\infty} \mathrm{r}_k^{(m)} = \mathrm{r}_k^{\mathsf{H}} \quad \forall k \geq 1.$$

A covariance stationary process whose autocorrelation function decays hyperbolically as in (1) is asymptotically second-order self-similar.

The most commonly used self-similar processes are Fractional Gaussian Noise (FGN), Fractional AutoRegressive Integrated Moving Average (F-ARIMA) and M/G/∞. The main disadvantage of FGN is that its correlation structure, determined by a single parameter, is too rigid to capture both the short-term and the long-term correlations simultaneously. Instead, F-ARIMA and M/G/∞-based processes are much more flexible for traffic modeling purposes.

3 F-ARIMA Processes

Fractional AutoRegressive Integrated Moving Average processes are a generalization of ARIMA processes.

A F-ARIMA($\mathsf{p}, \mathsf{d}, \mathsf{q}$) process $X = \{X_n; n = 1, 2, \dots\}$ satisfies the equation:

$$\phi_{\mathsf{p}}(B)(1-B)^{\mathsf{d}} X = \theta_{\mathsf{q}}(B)\epsilon,$$

where B is the backshift operator ($B^j X_n = X_{n-j}$), $\phi_{\mathsf{p}}(B)$ is a polynomial of order p in B, $\theta_{\mathsf{q}}(B)$ is a polynomial of order q in B, d is a real value and ϵ is a renewal process with zero mean and finite variance σ_ϵ^2.

Although it is not generally feasible to obtain the autocorrelation function for a F-ARIMA($\mathsf{p}, \mathsf{d}, \mathsf{q}$), for a F-ARIMA($0, \mathsf{d}, 0$) process is of the form:

$$\mathrm{r}_k = \frac{\Gamma(1-\mathsf{d})}{\Gamma(\mathsf{d})} \frac{\Gamma(k+\mathsf{d})}{\Gamma(k+1-\mathsf{d})} = \frac{\prod_{i=1}^{k}(\mathsf{d}+i-1)}{\prod_{i=1}^{k}(i-\mathsf{d})},$$

whose asymptotic behavior is:

$$r_k \sim \frac{\Gamma(1-\mathsf{d})}{\Gamma(\mathsf{d})} k^{2\mathsf{d}-1}.$$

So, for $0 < \mathsf{d} < 0.5$, the autocorrelation function exhibits a hyperbolic decay as expressed in (1) and the process is asymptotically second-order self-similar with $\mathsf{H} = \mathsf{d} + 0.5$.

Cox [7] extended this result, showing that any F-ARIMA$(\mathsf{p}, \mathsf{d}, \mathsf{q})$ process, with $0 < \mathsf{d} < 0.5$, is asymptotically second-order self-similar.

3.1 Synthetic Generation of F-ARIMA Processes

Although several methods exist for the generation of F-ARIMA processes, two must be highlighted:

- The exact method proposed by Hosking [17]. Although it is an exact method its main disadvantage is its extremely high computational cost, due to the calculation of each sample depends on all the previous ones. That makes it prohibitive for very long samples.
- The approximate method proposed by Ardao [24] as an extension of the method of Paxson for FGN. It is a very efficient method that generates high quality traces. Its main disadvantage is that all samples must be generated simultaneously, with the problems for simulation studies that this supposes:
  - the time of simulation is limited by the size of the traces obtained previously,
  - the size of the traces is limited by the available memory.

Other approximate methods for generating samples of the F-ARIMA processes are the method of Davies-Harte [9], the method of Haslett-Raftery [15] and the method based on the aggregation of AR(1) processes [13]. The advantages and drawbacks of each one are discussed in [24].

4 M/G/∞-Based Processes

The M/G/∞ process [6], X, is a stationary version of the occupancy process of an M/G/∞ queueing system. Let λ be the arrival rate to the system, and denote by S the service time distribution, with finite mean value $\mathsf{E}[S]$.

Although it is possible to use two different approximations to characterize the M/G/∞ process, a discrete time analysis [7] and a continuous time analysis [24], the most efficient way to generate it is to simulate the queue in discrete time, as it is exposed in [33].

If the initial number of users is a Poisson random variable of mean value $\lambda\mathsf{E}[S]$, and their service times are mutually independent and have the same distribution as the residual life of S, $\widehat{S}$:

$$\Pr\left[\widehat{S} = k\right] = \frac{\Pr[S >= k]}{\mathsf{E}[S]},$$

then the stochastic process X is strict-sense stationary, ergodic, and enjoys the following properties:

- the marginal distribution is Poissonian, with mean value: $\mathsf{E}[X] = \lambda \mathsf{E}[S]$,
- the autocorrelation function is:

$$r[k] = \Pr\left[\widehat{S} > k\right] \quad \forall k.$$

So, the autocorrelation structure of X is completely determined by the distribution of S.

In particular, the M/G/∞ process exhibits LRD when S has infinite variance, as it happens in heavy-tailed distributions.

In [20] the authors show that an $\Re^+$-valued sequence $r[k]$ can be the autocorrelation function of the stationary M/G/∞ process, with integrable S, if and only if it is decreasing and integer-convex, with $r[0] = 1$ and $\lim_{k\to\infty} r[k] = 0$, in which case the distribution function of S is given by:

$$\Pr[S \leq k] = 1 - \frac{r[k] - r[k+1]}{1 - r[1]} \quad \forall k > 0. \tag{3}$$

Its mean value is: $\mathsf{E}[S] = (1 - r[1])^{-1}$.

4.1 Efficient Generation of Synthetic Traces

In this section we explain briefly a method that we have proposed in previous works [31,32] in order to get a flexible and highly efficient M/G/∞ generator, based on the decomposition property of the Poisson processes and the memoryless property of the geometric random variables, and applicable to any distribution for the service time with subexponential decay.

When we use a discrete-time simulation model of the M/G/∞ system, every sample value of the occupancy process, X_n, requires the generation of one sample of the Poisson random variable A_n, with mean value λ, and A_n samples of the random variable S. We denote by M the mean number of random values that have to be generated for each sample value of X_n. In this case $\mathsf{M} = \lambda + 1$. For large values of λ, the computational time can be very high.

In order to reduce M, we divide the arrivals to the M/G/∞ system at each instant n into $\mathsf{K} + \mathsf{L} + 1$ groups, according to the random variable from which their service times are generated.

For the K first groups, the mean number of arrivals at each group is $\lambda \mathsf{d}_i; i = 1, 2, \ldots, \mathsf{K}$, being $\mathsf{d}_i = \Pr[S = i]; i = 1, 2, \ldots, \mathsf{K}$, and the service times are deterministic, with values $i = 1, 2, \ldots, \mathsf{K}$.

For the L following groups, we fit the distribution of the S random variable with the composition of the distributions of L geometric random variables, with parameters $\mathsf{p}_i; i = 1, 2, \ldots, \mathsf{L}$ and shifted to $k = \mathsf{K}_G = \mathsf{K} + 1$, $G_i^{'} = \mathsf{K}_G + G_i; i = 1, 2, \ldots, \mathsf{L}$. We denote by $\mathsf{g}_i^{'}; i = 1, 2, \ldots, \mathsf{L}$ the composition factors.

Finally, we denote by R the random variable whose distribution is:

$$\mathsf{r}^{-1}\left\{\Pr[S = k] - \sum_{i=1}^{\mathsf{L}} \mathsf{g}_i^{'} \Pr\left[G_i^{'} = k\right]\right\} \quad \forall k > \mathsf{K},$$

where:

$$\mathsf{r} = 1 - \sum_{i=1}^{\mathsf{K}} \mathsf{d}_i - \sum_{j=1}^{\mathsf{L}} \mathsf{g}_j^{'}$$

is the probability that a service time is generated from this random variable.

With this method, the mean number of random values that have to be generated for each sample value of the occupancy process is:

$$\mathsf{M} = \mathsf{K} + 2\mathsf{L} + 1 + \lambda \mathsf{r} = \mathsf{K} + 2\mathsf{L} + 1 + \lambda \left(1 - \sum_{i=1}^{\mathsf{K}} \mathsf{d}_i - \sum_{j=1}^{\mathsf{L}} \mathsf{g}_j^{'} \right).$$

In order to generate samples of R, we use a modified version of the tabular method of inversion of the cumulative distribution function of a non-negative discrete random variable proposed in [33].

The interval [0,1] is discretized into as many subintervals as the range of a pseudorandom number generator and then four tables are used to invert the distribution function.

Two of the tables, `index` and `DF`, support a binary search algorithm for the middle zone of the distribution.

The other two tables, `DF_inv_left` and `DF_inv_right`, are used to tabulate the values of the left and right tails of the distribution function, respectively. This is done to avoid losing precision in the generation of samples of random variables which show non-negligible probability mass at their extremes.

5 Whittle Estimator

Let $f_\theta(\lambda)$ be the spectral density function of a zero-mean Gaussian stochastic process, $X = \{X_n; n = 1, 2, \dots\}$ and let $I_{X^{\mathsf{N}}}(\lambda) = \frac{1}{2\pi\mathsf{N}} \left| \sum_{i=0}^{\mathsf{N}-1} X_{i+1} \mathrm{e}^{-\jmath\lambda i} \right|^2$ be the periodogram of a sample of size N of X. $\theta = \{\theta_k; k = 1, \dots, \mathsf{M}\}$ is the vector of parameters to be estimated.

The approximate Whittle MLE is the vector $\widehat{\theta} = \left\{ \widehat{\theta_k}; k = 1, \dots, \mathsf{M} \right\}$ that minimizes, for a given sample X^{N} of size N of X, the statistic:

$$Q_{X^{\mathsf{N}}}(\theta) \stackrel{\Delta}{=} \frac{1}{2\pi} \left[\int_{-\pi}^{\pi} \frac{I_{X^{\mathsf{N}}}(\lambda)}{f_\theta(\lambda)} \mathrm{d}\lambda + \int_{-\pi}^{\pi} \log f_\theta(\lambda) \, \mathrm{d}\lambda \right]. \tag{4}$$

Moreover, if θ^{o} is the real value of θ:

$$\Pr\left[\left| \hat{\theta} - \theta^{\mathrm{o}} \right| < \epsilon \right] \xrightarrow[\mathsf{N}\to\infty]{} 1 \quad \forall \epsilon > 0,$$

then $\sqrt{\mathsf{N}}(\hat{\theta} - \theta^{\mathrm{o}})$ converges in distribution to ζ, as $\mathsf{N} \to \infty$, where ζ is a zero-mean Gaussian vector with matrix of covariances $\mathrm{C}(\theta^{\mathrm{o}}) = 2\,\mathrm{D}^{-1}(\theta^{\mathrm{o}})$, being:

$$\mathrm{D}_{ij}(\theta^{\mathrm{o}}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\partial}{\partial\theta_i} \log f_\theta(\lambda) \frac{\partial}{\partial\theta_j} \log f_\theta(\lambda) \, \mathrm{d}\lambda \Big|_{\theta=\theta^{\mathrm{o}}}. \tag{5}$$

So, confidence intervals of the estimated values can be obtained.

A simplification of (4) can be achieved by choosing a special scale parameter θ_1, such that:

$$f_\theta(\lambda) = \theta_1 f_{\theta^*}(\lambda) = \theta_1 f^*_\eta(\lambda),$$

and:

$$\int_{-\pi}^{\pi} \log f_{\theta^*}(\lambda)\,\mathrm{d}\lambda = \int_{-\pi}^{\pi} \log f^*_\eta(\lambda)\,\mathrm{d}\lambda = 0,$$

where $\eta = \{\theta_i; i = 1, \ldots, \mathsf{M}\}$ and $\theta^* = (1, \eta)$.

Thus:

$$\theta_1 = \exp\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} \log f_\theta(\lambda)\,\mathrm{d}\lambda\right) = \frac{\sigma_\epsilon^2}{2\pi},$$

where σ_ϵ^2 is the optimal one-step-ahead prediction error, that is equal to the variance of the innovations of the AR(∞) representation of the process [2]:

$$X_i = \sum_{j=1}^{\infty} \beta_j X_{i-j} + \epsilon_i.$$

Equation (4) therefore simplifies to:

$$Q_{X^{\mathbb{N}}}(\theta^*) = Q^*_{X^{\mathbb{N}}}(\eta) = \int_{-\pi}^{\pi} \frac{I_{X^{\mathbb{N}}}(\lambda)}{f_{\theta^*}(\lambda)}\mathrm{d}\lambda = \int_{-\pi}^{\pi} \frac{I_{X^{\mathbb{N}}}(\lambda)}{f^*_\eta(\lambda)}\mathrm{d}\lambda.$$

Additionally [2]:

$$\widehat{\sigma_\epsilon^2} = Q^*_{X^{\mathbb{N}}}(\widehat{\eta}) \quad .$$

6 M/G/∞-Based Generation of the Autocorrelation Structure of F-ARIMA Processes

First, we consider the autocorrelation function of the F-ARIMA(0,d,0) process, i.e., of a fractional integration process:

$$r[k] = r_{\mathsf{d}}[k] \triangleq \frac{\prod_{i=1}^{k}(\mathsf{d}+i-1)}{\prod_{i=1}^{k}(i-\mathsf{d})}.$$

We denote by I the random variable for the service time in a M/G/∞ system generating an occupancy process with such correlation structure. From (3) the distribution function results:

$$\Pr[S \le k] = 1 - \frac{\mathrm{r}_k^{\mathsf{d}} - \mathrm{r}_{k+1}^{\mathsf{d}}}{1 - \dfrac{\mathsf{d}}{1-\mathsf{d}}} \quad \forall k > 0.$$

and the mean value:

$$\mathsf{E}[S] = \frac{1}{1 - \dfrac{\mathsf{d}}{1-\mathsf{d}}}.$$

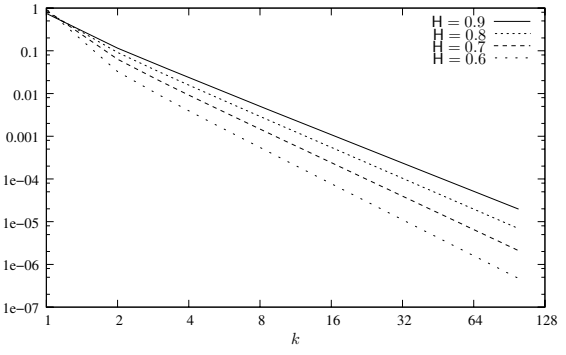


Fig. 1. Probability mass function of I

The distribution function of the residual life, needed in order to initialize the process in steady state is:

$$\Pr\left[\widehat{S} \leq k\right] = 1 - \mathrm{r}_k^{\mathsf{d}} \quad \forall k > 0.$$

In Fig. 1 we show the probability mass function of the service time for several values of the parameter H. We can observe that it has subexponential decay, so we can apply the method described in 4.1.

To improve the adjustment of the short-term correlation we propose to add an AutoRegressive (AR) filter.

If Y is a process of type M/I/∞ the new process is obtained as:

$$X_n = \alpha_1 X_{n-1} + \ldots + \alpha_{\mathsf{p}} X_{n-\mathsf{p}} + Y_n. \tag{6}$$

Using the backshift operator B the equation (6) can be expressed as:

$$Y_n = \phi_{\mathsf{p}}(B) X_n,$$

where $\phi_{\mathsf{p}}(B)$ is is a polynomial of order p in B:

$$\phi_{\mathsf{p}}(B) = 1 - \alpha_1 B - \ldots - \alpha_{\mathsf{p}} B^{\mathsf{p}} = 1 - \sum_{i=1}^{\mathsf{p}} \alpha_i B^i.$$

We denote the resulting process as M/I/∞-AR. In Fig. 2 we show the relationship between the two processes.

If the resulting autocorrelation function is decreasing and convex, we can obtain the distribution of the service time of the process X by means of (3). In other case, we will have to use an AR filter to generate X from Y, discarding a sufficient number of samples in order to initialize the generator approximately in steady state.

Specifically, we will focus on the particular case of an AR(1) filter. In this case, the mean values and covariances are related by:

$$\mathsf{E}[X] = \frac{\mathsf{E}[Y]}{1 - \alpha_1},$$

$$\gamma_k^X = \frac{1}{1-\alpha_1^2}\left(\gamma_k^Y + \sum_{i=1}^{\infty}\gamma_{k+i}^Y\alpha_1^i + \sum_{i=1}^{\infty}\gamma_{k-i}^Y\alpha_1^i\right),$$

and the autocorrelation function of X is the one of the F-ARIMA$(1, \mathsf{d}, 0)$ process.

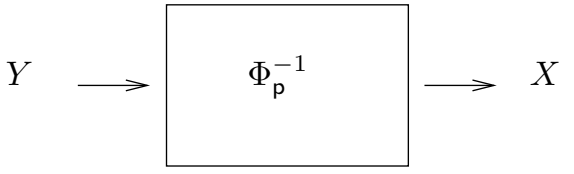


Fig. 2. Relationship between M/I/∞ and M/I/∞-AR

6.1 Efficiency of the Generator and Quality of the Traces

We have compared the efficiency of our method (METHOD 1) with that of the method of Ardao (METHOD 2), being the results obtained similar in both cases. In Table 1 we can see as an example the results (in seconds) for different lengths of the synthetic traces and different values of the Hurst parameter, considering the autocorrelation structure of F-ARIMA$(0, \mathsf{d}, 0)$ processes.

Table 1. Efficiency. Autocorrelation of F-ARIMA$(0, \mathsf{d}, 0)$.

N	H	METHOD 1	METHOD 2
10^6	0.6	5	3
10^6	0.75	6	4
10^6	0.9	6	4
10^7	0.6	48	38
10^7	0.75	53	43
10^7	0.9	57	48

Table 2. Quality. Autocorrelation of F-ARIMA$(0, \mathsf{d}, 0)$.

N	H	$\widehat{\mathsf{H}}$
10^6	0.6	0.599944
10^6	0.75	0.749764
10^6	0.9	0.901234
10^7	0.6	0.600116
10^7	0.75	0.750076
10^7	0.9	0.900041

And related to the quality of the samples, in Table 2 we can see the estimated parameter $\widehat{\mathsf{H}}$ using the Whittle estimator and the method of independent replies, for different values of N and H, considering the autocorrelation structure of F-ARIMA$(0, \mathsf{d}, 0)$ processes.

7 Application: Modeling of the Correlation Structure of VBR Video Traffic

In this section we show that F-ARIMA processes are more flexible in order to fit the autocorrelation function obtained from some empirical VBR traces that the FGN process.

We consider the following empirical trace of the Group of Pictures (GoP) sizes of the MPEG encoded video *"The Lord of the Rings"*, with length $\mathsf{N} = 66000$. We have generated it from the three parts of the trilogy.

In order to use the Whittle estimator to estimate the parameters of each process, we need the parametric form of its spectral density function.

For the FGN process the spectral density:

$$f(\lambda) = f_{\mathsf{H}}(\lambda) \stackrel{\Delta}{=} \mathsf{c}_f \left|\mathsf{e}^{j\lambda} - 1\right|^2 \sum_{i=-\infty}^{+\infty} |2\pi i + \lambda|^{-2\mathsf{H}-1} \quad \forall \lambda \in [-\pi, \pi],$$

can be computed efficiently with the Euler's formula.

For the F-ARIMA$(0, \mathsf{d}, 0)$ process the spectral density is:

$$f(\lambda) = f_{\mathsf{d}}(\lambda) \stackrel{\Delta}{=} \frac{\sigma_\epsilon^2}{2\pi} |1 - \mathsf{e}^{j\lambda}|^{-2\mathsf{d}} = \frac{\sigma_\epsilon^2}{2\pi} \left(2 \sin\left(\frac{\lambda}{2}\right)\right) \quad \forall \lambda \in [-\pi, \pi].$$

For the F-ARIMA$(1, \mathsf{d}, 0)$ process the spectral density is:

$$f_X(\lambda) = \frac{f_Y(\lambda)}{|1 - \alpha_1 \mathsf{e}^{j\lambda}|^2} \quad \forall \lambda \in [-\pi, \pi].$$

In order to adjust simultaneously the marginal distribution and the autocorrelation, as the marginal distribution in all cases is approximately Lognormal, we apply a change of distribution.

After the transformation the estimations are as follows:

- FGN process: $\widehat{\mathsf{H}} = 0.741$.
- F-ARIMA$(0, \mathsf{d}, 0)$ process: $\widehat{\mathsf{H}} = 0.788$.
- F-ARIMA$(1, \mathsf{d}, 0)$ process: $\widehat{\alpha_1} = -0.086$ and $\widehat{\mathsf{H}} = 0.831$.

The variance (or confidence intervals) of the estimations can be computed from (5).

We use $\widehat{\sigma_\epsilon^2}$ as a measure of the suitability of each process, since smaller values of $\widehat{\sigma_\epsilon^2}$ mean numerically better adjustment to the empirical correlation of the sample.

In Table 3 we show the estimations of the prediction error.

The results show that the F-ARIMA$(1, \mathsf{d}, 0)$ process leads to smaller prediction errors.

As open questions in the use of this selection criterion, we may consider the following ones:

Table 3. $\widehat{\sigma_\epsilon^2}$ with each adjustment

FGN	$7.349 \cdot 10^{-2}$
F-ARIMA$(0, \mathsf{d}, 0)$	$7.277 \cdot 10^{-2}$
F-ARIMA$(1, \mathsf{d}, 0)$	$7.251 \cdot 10^{-2}$

- Being the difference between the respective values of σ_ϵ^2 of each two models so small, is this difference significant?
- F-ARIMA$(1, \mathsf{d}, 0)$ supposes a major flexibility in the adjustment of the autocorrelation function that F-ARIMA$(0, \mathsf{d}, 0)$, and therefore a reduction of the estimation of the prediction error, but is this improvement significant in order to compensate the increase of complexity of the model?

To solve these questions, in an further work we are going to propose an hypothesis test over the spectral density.

8 Conclusions and Further Work

In this paper we have proposed an efficient on-line generator of the correlation structure of F-ARIMA processes based on the M/G/∞ process. We have checked the efficiency and the quality of the traces, being the results obtained very satisfactory. With an example, we have shown numerically that F-ARIMA processes are more flexible in order to fit the autocorrelation function of some empirical traces that the FGN processes. In an extended version of this paper we are going to study if the numerically better adjustment is significant or not, by means of an hypothesis test over the spectral density.

References

1. Ansari, N., Liu, H., Shi, Y.Q.: On modeling MPEG video traffics. IEEE Transactions on Broadcasting 48(4), 337–347 (2002)
2. Beran, J.: Statistics for Long-Memory Processes. Chapman and Hall, Boca Raton (1994)
3. Beran, J., Shreman, R., Taqqu, M.S., Willinger, W.: Long-Range Dependence in Variable-Bit-Rate video traffic. IEEE Transactions on Communications 43(2/4), 1566–1579 (1995)
4. Casilari, E., Reyes, A., Díaz, A., Sandoval, F.: Characterization and modeling of VBR video traffic. Electronics Letters 34(10), 968–969 (1998)
5. Conti, M., Gregori, E., Larsson, A.: Study of the impact of MPEG-1 correlations on video sources statistical multiplexing. IEEE Journal on Selected Areas in Communications 14(7), 1455–1471 (1996)
6. Cox, D.R., Isham, V.: Point Processes. Chapman and Hall, Boca Raton (1980)
7. Cox, D.R.: Long-Range Dependence: A review. In: Statistics: An Appraisal, pp. 55–74. Iowa State University Press (1984)
8. Crovella, M.E., Bestavros, A.: Self-similarity in World Wide Web traffic: Evidence and possible causes. IEEE/ACM Transactions on Networking 5(6), 835–846 (1997)

9. Davies, R.B., Harte, D.S.: Tests for Hurst effect. Biometrika 74(1), 95–102 (1987)
10. Duffield, N.: Queueing at large resources driven by long-tailed M/G/∞ processes. Queueing Systems 28(1/3), 245–266 (1987)
11. Erramilli, A., Narayan, O., Willinger, W.: Experimental queueing analysis with Long-Range Dependent packet traffic. IEEE/ACM Transactions on Networking 4(2), 209–223 (1996)
12. Garrett, M.W., Willinger, W.: Analysis, modeling and generation of self-similar VBR video traffic. In: Proc. ACM SIGCOMM 1994, London, UK, pp. 269–280 (1994)
13. Granger, C.W.J.: Long memory relationships and the aggregation of dynamic models. Journal of Econometrics 14(2), 227–238 (1980)
14. Granger, C.W.J., Joyeux, R.: An introduction to long-range time series models and fractional differencing. Journal of Time Series Analysis 1, 15–30 (1980)
15. Haslett, J., Raftery, A.E.: Space-time modeling with long-memory dependence: Assessing Ireland's wind power resource. Applied Statistics 38(1), 1–50 (1989)
16. Hosking, J.R.M.: Fractional differencing. Biometrika 68(1), 165–176 (1981)
17. Hosking, J.R.M.: Modeling persistence in hydrological time series using fractional differencing. Water Resources Research 20(12), 1898–1908 (1984)
18. Hurst, H.E.: Long-term storage capacity of reservoirs. Transactions of the American Society of Civil Engineers 116, 770–799 (1951)
19. Jiang, M., Nikolic, M., Hardy, S., Trajkovic, L.: Impact of self-similarity on wireless data network performance. In: Proc. IEEE ICC 2001, Helsinki, Finland, pp. 477–481 (2001)
20. Krunz, M., Makowski, A.: Modeling video traffic using M/G/∞ input processes: A compromise between Markovian and LRD models. IEEE Journal on Selected Areas in Communications 16(5), 733–748 (1998)
21. Leland, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V.: On the self-similar nature of Ethernet traffic (extended version). IEEE/ACM Transactions on Networking 2(1), 1–15 (1994)
22. Li, S.Q., Hwang, C.L.: Queue response to input correlation functions: Discrete spectral analysis. IEEE/ACM Transactions on Networking 1(5), 317–329 (1993)
23. Likhanov, N., Tsybakov, B., Georganas, N.D.: Analysis of an ATM buffer with self-similar ("fractal") input traffic. In: Proc. IEEE INFOCOM 1995, Boston, MA, USA, pp. 985–992 (1995)
24. López, J.C., López, C., Suárez, A., Fernández, M., Rodríguez, R.F.: On the use of self-similar processes in network simulation. ACM Transactions on Modeling and Computer Simulation 10(2), 125–151 (2000)
25. Mandelbrot, B.B., Van Ness, J.W.: Fractional Brownian Motions, Fractional Noises and applications. SIAM Review 10(4), 422–437 (1968)
26. Norros, I.: A storage model with self-similar input. Queueing Systems 16, 387–396 (1994)
27. Parulekar, M.: Buffer engineering for M/G/∞ input processes. Ph.D. Thesis, University of Maryland, College Park, MD, USA (2001)
28. Paxson, V., Floyd, S.: Wide-area traffic: The failure of Poisson modeling. IEEE/ACM Transactions on Networking 3(3), 226–244 (1995)
29. Poon, W., Lo, K.: A refined version of M/G/∞ processes for modeling VBR video traffic. Computer Communications 24(11), 1105–1114 (2001)
30. Resnick, S., Rootzen, H.: Self-similar communication models and very heavy tails. Annals of Applied Probability 10(3), 753–778 (2000)

31. Sousa, M.E., Suárez, A., Fernández, M., López, C., Rodríguez, R.F.: A highly efficient M/G/∞ generator of self-similar traces. In: Proc. 2006 Winter Simulation Conference, Monterey, CA, USA, pp. 2146–2153 (2006)
32. Sousa, M.E., Suárez, A., López, J.C., López, C., Fernández, M.: On improving the efficiency of a M/G/∞ generator of correlated traces. Operations Research Letters 36(2), 184–188 (2008)
33. Suárez, A., López, J.C., López, C., Fernández, M., Rodríguez, R.F., Sousa, M.E.: A new heavy-tailed discrete distribution for LRD M/G/∞ sample generation. Performance Evaluation 47(2/3), 197–219 (2002)
34. Taqqu, M.S., Teverovsky, V.: On estimating the intensity of Long-Range Dependence in finite and infinite variance time series. In: A Practical Guide to Heavy Tails, pp. 177–218. Birkhauser, Basel (1998)
35. Tsoukatos, K.P., Makowski, A.M.: Heavy traffic analysis for a multiplexer driven by M/G/∞ input processes. In: Proc. 15th International Teletraffic Congress, Washington, DC, USA, pp. 497–506 (1997)
36. Whittle, P.: Estimation and information in stationary time series. Arkiv. Matematick 2(23), 423–434 (1953)

Different Monotonicity Definitions in Stochastic Modelling*

Imène Kadi[1], Nihal Pekergin[2], and Jean-Marc Vincent[3]

[1] PRiSM, University Versailles-Saint-Quentin, 45 Av. des Etats-Unis 78000 France
[2] LACL, University Paris-Est, 61 avenue Général de Gaulle 94010, Créteil, France
[3] LIG, project-INRIA MESCAL, 51, av. Jean Kuntzmann,
38330 Montbonnot, France
imene.kadi@prism.uvsq.fr,
nihal.pekergin@univ-paris12.fr,
jean-marc.vincent@imag.fr

Abstract. In this paper we discuss different monotonicity definitions applied in stochastic modelling. Obviously, the relationships between the monotonicity concepts depends on the relation order that we consider on the state space. In the case of total ordering, the stochastic monotonicity used to build bounding models and the realizable monotonicity used in perfect simulation are equivalent to each other while in the case of partial order there is only implication between them. Indeed, there are cases of partial order, where we can't move from the stochastic monotonicity to the realizable monotonicity, this is why we will try to find the conditions for which there are equivalences between these two notions. In this study, we will present some examples to give better intuition and explanation of these concepts.

1 Introduction

Simulation approaches constitute an alternative for performance evaluation, when numerical methods fail. In fact, they are usually used to model complex systems, such as, optical networks, distributed computer systems, stochastic Petri networks, and so on. In this paper we advocate the use of perfect simulation and combining this technique with stochastic monotonicity to speed up the computation. This method is based on the more general theory of coupling for Markov chains. Let us first review some ideas about coupling. Assume that we compute with the same random sequence of random numbers a sample path beginning at any initial state. If at time t two sample-paths are in the same state (we say that they couple), they will continue forever during all the simulation. When all the sample-paths have coupled, we obtain a sample state. We may use the state to initialize the simulation or consider it as a sample, thus it is not necessary anymore to continue the simulation.

It is known for a long time that coupling in the future does not provide samples distributed according to the steady-state distribution. But Propp and

* Partially supported by french projects ANR-Blanc SMS, ANR- SETi06-02.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 144–158, 2009.

Wilson have proved that coupling from the past (CFTP), also called backward-coupling, gives an exact sample of the steady-state distribution [12]. Coupling from the past is similar to coupling in the future but the initial time of the simulation will be chosen randomly whereas the final time is deterministic. In other words the Markov chain is not started at time 0 but sufficiently far away in the past such that at time 0 all the paths are coupled.

This method is extremely efficient. But many practical and theoretical problems remain to be solved for discrete Markovian systems to obtain a fully versatile technique. One of the problem we must consider is the number of operations we need to obtain a sample. The general backward algorithm tries to couple sample-paths beginning in every state in the state space. Thus modelling very large state space systems requires some model transformations. Furthermore the number of operations is at least linear in the size of the state space. The monotonicity property of the event structure of the model (which is formally defined in the next section) allows us to use a more efficient algorithm which sandwiches all sample-paths to couple into extreme ones.

We consider in this paper different monotonicity definitions applied in different context of stochastic modelling. First of them is the stochastic monotonicity concept associated to a stochastic ordering relation. This implies that the evolution of the underlying model is monotone regarding to the considered stochastic order. This monotonicity concept is one of the sufficient conditions to build bounding models [14]. For performability analysis of complex models, bounding models rather than the original one are considered to verify if performability requirements are satisfied by the original model. Obviously the bounding models must be easier to analyze than the original one [7].

In general the considered order relation on the state space is a total ordering. However the partial order is more suitable for multidimensional models. We explain first the stochastic monotonicity for a state space endowed with at least a pre-order and study the relationships with other monotonicity definitions.

The remaining monotonicity definitions are related to perfect simulation (sandwiching property). The first concept is called realizable monotonicity and was defined in [4]. The other definition is used in a software to provide perfect simulation of queueing networks (http://www-id.imag.fr/Logiciel/psi/). This is called event monotonicity and has been defined in more general terms in [8].

In this paper we present these definitions by emphasizing if the state space is totally ordered or not. We then compare them to give insights for the implications between them. We have considered relations between monotonicity definitions in a totally ordered state space [6]. In this case, the stochastic monotonicity and the event monotonicity are equivalent to each other. Therefore it is possible to construct bounding and stochastic monotone models in order to do monotone perfect simulations of systems which are not event monotone.

This paper is organized as follows: The next section is devoted to a brief presentation of considered stochastic models, perfect simulation, and stochastic ordering. In section 3, we give the different definitions of monotonicity: first the monotonicity in the sense of strong stochastic ordering then the realizable and

event monotonicity used in perfect simulation. We present monotone perfect simulation of realizable monotone models in section 4. In section 5, we study the relationships between the stochastic monotonicity and the realizable monotonicity in order to see if stochastic monotone models can be used to perform monotone perfect simulation. So we show that these notions are different in the case of a totally and partially ordered state spaces. This is why we try to find a case of equivalence of these two notions under a partial order, and we give algorithms to construct event monotone systems in these cases.

2 Preliminaries

Markovian Discrete Event Systems (MDES) are dynamic systems evolving asynchronously and interacting at irregular instants called *event epochs* [8]. These systems are defined by means of a state space $\mathcal{X}$, a set of events $\mathcal{E}$, a set of probability measures $\mathcal{P}$, and a transition function Φ. $\mathbb{P}(e) \in \mathcal{P}$ denotes the occurrence probability of event $e \in \mathcal{E}$ while $\Phi(x, e)$ denotes the state to which the system moves from state x upon the occurrence of an event $e \in \mathcal{E}$.

Definition 1 (event). *An event e is an application defined on $\mathcal{X}$, that associates to each state $x \in \mathcal{X}$ a new state $y \in \mathcal{X}$.*

Definition 2 (Transition function). *Let X_i be the state of the system at the i^{th} event occurrence time. The transition function $\Phi : \mathcal{X} \times \mathcal{E} \rightarrow \mathcal{X}$, defines the next state of the system X_{n+1} resulting from X_n upon the occurrence of an event e_{n+1}:*

$$X_{n+1} = \Phi(X_n, e_{n+1}) \tag{1}$$

Φ must to obey to the following property to generate $\mathbf{P}$:

$$p_{ij} = \mathbb{P}(\phi(x_i, E) = x_j) = \sum_{e|\Phi(x_i,e)=x_j} \mathbb{P}(E = e) \tag{2}$$

Markov processes constitute a special, perhaps the most important subclass of stochastic processes [1]. We restrict ourselves here to the investigation of discrete state space and in that case refer to the stochastic processes as *chains*. Discrete Time Markov Chains(DTMC) are considered first, that is, Markov processes restricted to discrete, finite, or countably infinite state space, $\mathcal{X}$, and a discrete-parameter space T(time). For the sake of convenience, we set $T \subseteq \mathbb{N}_0$.

We consider in this work only time-homogeneous Markov chains, i.e, the conditional distribution function of a state X_{n+1} does not depend on observation time, that is, it is invariant with respect to time epochs n.

Definition 3 (DTMC). *A given stochastic process $\{X_0, X_1, ..., X_{n+1}, ...\}$ at the consecutive points of observation $0, 1, ..., n+1$ constitutes a DTMC if the following relation on the conditional probability mass function(pmf), that is, the Markov property, holds for all $n \in \mathbb{N}$ and all $x_i \in \mathcal{X}$:*

$$\mathbb{P}(X_{n+1}=x_{n+1}|X_n=x_n, X_{n-1}=x_{n-1}, ..., X_0=x_0)=\mathbb{P}(X_{n+1}=x_{n+1}|X_n=x_n). \tag{3}$$

Let $\mathcal{X} = \{0, 1, 2, ...\}$ and write conveniently the notation for the conditional pmf of the process's one-step transition from state i to state j at time n:

$$p_{ij}(n) = \mathbb{P}(X_{n+1} = x_{n+1} = j | X_n = x_n = i). \tag{4}$$

The one-step transition probability p_{ij} are given in a non-negative, stochastic[1] transition matrix $\mathbf{P}$:

$$\mathbf{P} = \mathbf{P}^{(1)} = [p_{ij}] \begin{pmatrix} p_{00} & p_{01} & p_{02} & \cdots \\ p_{10} & p_{11} & p_{12} & \cdots \\ p_{20} & p_{21} & p_{22} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

The following proposition gives how we can construct a transition function Φ for a time-homogeneous DTMC with a probability transition matrix $\mathbf{P}$ [9].

Definition 4. *A probability transition matrix $\mathbf{P}$, on a partially ordered state space $(\mathcal{X}, \preceq)$, can be described by a transition function*

$\Phi : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$, where U is a random variable taking values in an arbitrary probability space $\mathcal{U}$, such that, for all $x, y \in \mathcal{X}$: $\mathbb{P}(\Phi(x, U) = y) = p_{xy}$:

$$X_{n+1} = \Phi(X_n, U_{n+1}) \tag{5}$$

2.1 Perfect Sampling

Based on the transition function Φ, the following algorithm provides directly a sample of the steady state distribution. Let $\mathcal{X}$ be finite state space set.

Algorithm 1. Backward coupling simulation

```
n=1;
E[1]=Generate-event();
repeat
   n=2.n;
   for all x ∈ X do
      Y[x] ← x; {initialization of trajectories, size of vector Y is |X|}
   end for
   for i=n downto n/2+1 do
      E[i]=Generate-event(); {generation of new events from -n/2 +1 to -n}
   end for
   for i=n downto 1 do
      Y ← Φ(Y, E[i]); {generation of trajectories through events E[i], }
   end for
   {Y[x] is the state reached at time 0 for the trajectory issued from x at time -n}
until All Y[x] are equal; {Coupling of all trajectories at time 0}
```

[1] The elements in each row of the matrix sum up to1.

Let $\mathbb{E}\tau$ be the expectation of the coupling time, $|\mathcal{X}|$ be the size of the state space and $op(\Phi)$ be the average number of operations to compute the transited state. Clearly the average number of operations before coupling is $|\mathcal{X}|.\mathbb{E}\tau.op(\Phi)$.

Function Φ has a lot of influence on the number of operations. First the way it is implemented has a linear influence because of term $op(\Phi)$.

2.2 Stochastic Ordering

Here we present the stochastic ordering of random variables and Markov chains. We refer to [14] for further informations. Let $\mathcal{X}$ be a discrete countable state space. We consider that $\mathcal{X}$ is endowed with at least a pre-order $\preceq$.The strong stochastic ordering associated to $\preceq$ will be denoted by $\preceq_{st}$.

A stochastic order can be defined by means of two approaches. The first way is to define them from a set of functions. The stochastic order defined in this case are called integral order. The second way is to define them from increasing sets which is more useful when the state space is not totally ordered.

Definition 5. *Let X and Y be two random variables taking values on $\mathcal{X}$.*

$$X \preceq_{st} Y \Leftrightarrow \mathbb{E}f(X) \leq \mathbb{E}f(Y)$$

for all function $f : \mathcal{X} \to R$ which is not decreasing according to relation $\preceq$ whenever the expectations exist.

When the state space is totally ordered, the above definition implies the following property:

Property 1. *Let X and Y be two random variables taking values on $\mathcal{X}$, with a total order $\leqslant$, and let F_X and F_Y be respectively their distribution functions:*

$$X \leqslant_{st} Y \Leftrightarrow F_X(a) \geq F_Y(a), \forall a \in \mathcal{X}$$

From the order relation (at least pre-order) $\preceq$ on $\mathcal{X}$, we can define increasing sets on $\mathcal{X}$.

Definition 6. *[Increasing set] Any subset Γ of $\mathcal{X}$ is called an increasing set if $x \preceq y$ and $x \in \Gamma$ implies $y \in \Gamma$.*

The stochastic order $\preceq_{st}$ is defined as follows from increasing sets:

Definition 7. *Let T and V be two discrete random variables and Γ an increasing set defined on $\mathcal{X}$*

$$T \preceq_{st} V \Leftrightarrow \sum_{x \in \Gamma} \mathbb{P}(T = x) \leq \sum_{x \in \Gamma} \mathbb{P}(V = x), \quad \forall \Gamma.$$

3 Different Definitions of Monotonicity

Here we present different monotonicity definitions used in stochastic modelling. First we give the stochastic monotonicity associated to the stochastic order $\preceq_{st}$ then give the monotonicity definitions used for the perfect simulation.

3.1 Stochastic Monotonicity

Following [14,10] let us give the definition of the stochastic monotonicity for probability transition matrices of time-homogeneous DTMCs.

Definition 8 (stochastic monotonicity). *Let* $\mathbf{P}$ *be a stochastic matrix,* $\mathbf{P}$ *is st-monotone if and only if for any probability vectors on* $\mathcal{X}$*,* u *and* v*, if* $u \preceq_{st} v$ *implies that* $u\mathbf{P} \preceq_{st} v\mathbf{P}$*.*

Definition 9. *Let* $\mathbf{P}$ *be the transition probability matrix of a time-homogeneous Markov chain* $\{X_n, n \geq 0\}$ *taking values in* $\mathcal{X}$ *endowed with relation order* $\preceq$*.* $\{X_n, n \geq 0\}$ *is st-monotone if and only if,*

$\forall (x, y) \mid x \preceq y$ *and* $\forall$ *increasing set* $\Gamma \in \mathcal{X}$

$$\sum_{z \in \Gamma} p_{xz} \leq \sum_{z \in \Gamma} p_{yz} \tag{6}$$

If the state space is totally ordered, the st-monotonicity implies that the rows of $\mathbf{P}$ are increasing:

Property 2. *In the case of totally ordered state spaces,* $\mathbf{P}$ *is st-monotone if and only if for all* i*, we have* $P_{i,*} \leqslant_{st} P_{i+1,*}$*.*

In the following example, we discuss the st-monotonicity by considering respectively a total order and then a partial order relation on the state space to show that there is no implication. Let us remark here that we consider partial orders compatible with the considered total order in the sense that the relation orders for the partial order exist also in the total order, but some states are not comparable under the partial order.

Example 1

$$\mathbf{P} = \begin{pmatrix} 1/2 & 1/6 & 1/3 & 0 \\ 1/2 & 1/6 & 0 & 1/3 \\ 1/2 & 0 & 1/6 & 1/3 \\ 0 & 0 & 2/3 & 1/3 \end{pmatrix}$$

First we consider a total order: $\mathcal{X} = \{a, b, c, d\}$ *and* $a \leqslant b \leqslant c \leqslant d$*. We can see easily that the rows are increasing (property. (2)), so the matrix is stochastic monotone in the total ordering. Now we consider a partial order:* $a \preceq b \preceq d$*; and* $a \preceq c \preceq d$*. The increasing sets are* $\Gamma_1 = \{d\}$*,* $\Gamma_2 = \{c,\ d\}$*,* $\Gamma_3 = \{b, d\}$*,* $\Gamma_4 = \{b,\ c,\ d\}$*,* $\Gamma_5 = \{a,\ b,\ c,\ d\}$*.* $\mathbf{P}$ *is not monotone with respect to this order. For instance, for* $\Gamma_3 = \{b, d\}$*, the probability measure for row* b *is* $1/6 + 1/3$*, while this measure is* $1/3$ *for row* d*. Since* $b \preceq d$*, this violates the monotonicity.*

Therefore we can see that the monotonicity with a total order does not imply the monotonicity with a partial order. From a first view, it may seem to be a contradiction, because with total order we must compare all of the rows, however with partial order we consider only comparable states. For example, we do not compare row b *and* c *for partial order in this example. However we do not have the same increasing sets for these cases, for instance* $\Gamma_3 = \{b, d\}$ *is not an increasing set with total order.*

Property 3. *If* $\mathbf{P}$ *is* $\leqslant_{st}$*-monotone with respect to a total order defined on* $\mathcal{X}$*, then* $\mathbf{P}$ *is not always* $\preceq_{st}$*-monotone with respect to a partial order defined on* $\mathcal{X}$*.*

3.2 Realizable Monotonicity

First, we will give the definition of realizable monotonicity, used in Fill and Machida's works on the perfect simulation [5].

Definition 10 (realizable monotonicity). *Let* $\mathbf{P}$ *be a stochastic matrix defined on state space* $\mathcal{X}$*.* $\mathbf{P}$ *is said to be realizable monotone, if there exists a transition function* Φ *as in Eq. 5, such that* Φ *preserves the order relation i.e. for all* $u \in \mathbf{U}$*, we have* $\Phi(x,u) \preceq \Phi(y,u)$*, whenever* $x \preceq y$*.*

There is an other definition of monotonicity used to perform perfect simulation of finite queuing networks by software Psi2 [15].

Definition 11 (event monotonicity). *The underlying model is said to be event monotone, if the transition function by events (Eq. 1) preserves the order ie. for each* $e \in \mathcal{E}$

$$\forall (x,y) \in \mathcal{X} \;\; x \preceq y \Longrightarrow \Phi(x,e) \preceq \Phi(y,e)$$

This notion of event monotonicity is the same as the realizable monotonicity if the set of events $\mathcal{E}$ is pre-defined. So a system is realizable monotone means that there exists a finite set of events $\mathcal{E}$ for which the system is event monotone. In the case of finite DTMCs, the cardinality of the set of events is upper bounded by the number of non null entries of the transition matrix.

Example 2. *Let* $(\mathcal{X}, \preceq)$ *be a partial ordering state space,* $\mathcal{X} = \{a,b,c,d\}$*,* $a \preceq b \preceq d$ *and* $a \preceq c \preceq d$ *;*

We consider three events with the following probabilities $p_{e_1} = 1/6$*,* $p_{e_2} = 1/3$*,* $p_{e_3} = 1/2$*.*

$$\mathbf{P} = \begin{pmatrix} 1/2 & 1/3 & 0 & 1/6 \\ 1/2 & 1/6 & 0 & 1/3 \\ 1/2 & 1/3 & 0 & 1/6 \\ 0 & 1/3 & 1/6 & 1/2 \end{pmatrix}$$

	1/6	1/6	1/6	1/6	1/6	1/6
a	a			b		d
b	a			b	d	
c	a			b		d
d	b		c	d		

	e_3	e_2	e_1
a	a	b	d
b	a	d	b
c	a	b	d
d	b	d	c

If we consider the initial set of event, we can see from the first table that $\mathbf{P}$ *is realizable monotone, but it is not event monotone, for instance we have in the second table, for event* $p_{e_1} = 1/6$ *,* $\Phi(b,e_1) = b$ *is incomparable with* $\Phi(d,e_1) = c$*.*

But if we change the set of events, and define new events following the first table, we obtain an event monotone system. For instance, we can, from the table of realizable monotonicity, divide the interval [0,1] into monotone events, we obtain five events with the following probabilities $p_{e_1} = 1/3$, $p_{e_2} = 1/6$, $p_{e_3} = 1/6$, $p_{e_4} = 1/6$, $p_{e_4} = 1/6$.

	p_{e_1}	p_{e_2}	p_{e_3}	p_{e_4}	p_{e_5}
a	a	a	b	b	d
b	a	a	b	d	d
c	a	a	b	b	d
d	b	c	d	d	d

We summarize the relationships between these types of monotonicity by the following scheme 1. We can see that there no implication between monotonicity under the total order and a partial order compatible with the total order neither for the stochastic monotonicity nor the realizable monotonicity. When the state space is totally ordered, both monotonicity notions are equivalent while for partially ordered state spaces the realizable monotonicity implies the stochastic monotonicity.

4 Realizable Monotonicity and Perfect Sampling

When the operator Φ is realizable monotone, the algorithm could be simplified by making iteration only on maximal and minimal values of the state space. If the trajectories issued from minimal and maximal states are coupled, due to the realizable monotonicity, trajectories issued from all other states are also coupled. The perfect simulation of monotone models will clearly reduce the computation and memory complexity to obtain a sample [15].

We give in the following backward-coupling for event monotone models. Let us turn now to the expectation of the coupling time for event-monotone systems. In the algorithm M (resp. m) denotes the set of maximal (resp. minimal) elements in the state space. This algorithm has the same convergence properties as Algorithm (1). Thus the expected number of operations is $(M + m).\mathbb{E}\tau_1.op(\Phi)$.

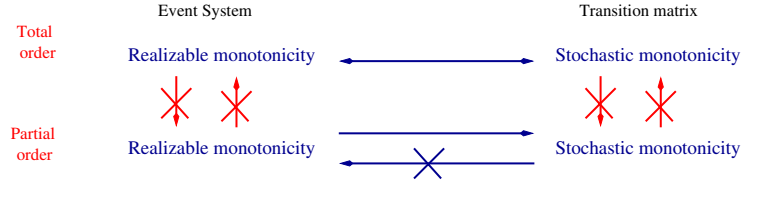


Fig. 1. Relations between monotonicity notions in total and partial order

Algorithm 2. Backward-coupling simulation (event monotone version)

```
1: n=1;
2: E[1]=Generate-event();
3: repeat
4:    n=2.n;
5:    for all x ∈ M ∪ m do
6:       Y[x] ← x; {initialization of trajectories,size of vector Y is |M ∪ m|}
7:    end for
8:    for i=t downto t/2+1 do
9:       E[i]=Generate-event(); {generation of new events from -n/2 +1 to -n}
10:   end for
11:   for i=n downto 1 do
12:      Y ← Φ(Y, E[i]); {generation of trajectories through events E[i], }
13:   end for{Y[x] is the state reached at time 0 for the trajectory issued from x at
      time -n}
14: until All Y([x] are equal; {Coupling of maximal and minimal trajectories at time
    0}
```

5 Stochastic Monotonicity and Perfect Simulation

Now we discuss how one can perform a monotone perfect simulation of a stochastic monotone DTMC. So we will study the relations between the stochastic monotonicity and the realizable monotonicity, and find the conditions that allow us to move from a stochastic monotone DTMC to an event monotone MDES.

5.1 Totally Ordered State Space

When the state space is totally ordered the stochastic monotonicity and the realizable monotonicity are equivalent [6]. However the stochastic monotonicity is necessary but not sufficient for realizable monotonicity for partially ordered state spaces. [4]

Theorem 1. *When the state space is totally ordered ($\leqslant$), the stochastic monotonicity and the realizable monotonicity are equivalent.*

This result has already been proved, but for better comprehension we will give a proof to this theorem.

Proof

– Realizable monotonicity $\Longrightarrow$ Stochastic monotonicity From the realizable monotonicity definition, we have for each two states x and $y \in \mathcal{X}$:

$$if\ x\ \leqslant\ y\ then\ \forall u \in [0,1] : \Phi(x,u) \leqslant \Phi(y,u) \tag{7}$$

The function Φ is the inverse probability distribution function. Let X and Y be two random variables corresponding respectively to rows x and y of $\mathbf{P}$. So $\Phi(x,u) = F_X^{-1}$, $\forall u$.

From (7), we obtain :

$$F_X^{-1}(u) \leqslant F_Y^{-1}(u), \forall u$$

this implies that for each state $a \in \mathcal{X}$:

$$F_X(a) \geq F_Y(a)$$

It follows from the definition of the strong stochastic ordering (property(1)) that $X \leqslant_{st} Y$. Thus, the model is stochastically monotone.

– Stochastic monotonicity $\Longrightarrow$ Realizable monotonicity
From the stochastic monotonicity, we have for each two states x and $y \in \mathcal{X}$:

$$if\ x\ \leqslant\ y\ then\ P[x,*] \preceq_{st} P[y,*] \tag{8}$$

Let X and Y be two random variables corresponding respectively to rows x and y of $\mathbf{P}$. From equation (8) and property(1) of strong stochastic ordering we obtain:

$$F_X(a) \geq F_Y(a), \forall a \in \mathcal{X} \tag{9}$$

Let u be a random variable, uniformally distributed in [0,1]. The equation (9) implies that :

$$\forall u \in [0,1] : F_X^{-1}(u) \leqslant F_Y^{-1}(u)$$

We see that the function F^{-1} satisfies the conditions of monotonicity. So we can always find a monotone transition function for the system.

Example 3. *Let* $\mathbf{P}3$ *be a transition matrix defined on a total ordered state space* $(\mathcal{X}, \leqslant)$, $\mathcal{X} = \{a, b, c, d\}$ *and* $a \leqslant b \leqslant c \leqslant d$.

$$\mathbf{P}3 = \begin{pmatrix} 1/2 & 1/6 & 1/3 & 0 \\ 1/2 & 1/6 & 0 & 1/3 \\ 1/2 & 0 & 1/6 & 1/3 \\ 0 & 1/6 & 1/2 & 1/3 \end{pmatrix}$$

It can be easily verified that $\mathbf{P}3$ *is st-monotone. This model can be described by a transition function* Φ*, obtained by the inverse probability distribution function.*

	1/6	*1/6*	*1/6*	*1/6*	*1/6*	*1/6*
a	*a*			*b*	*c*	
b	*a*			*b*	*d*	
c	*a*			*c*	*d*	
d	*b*			*c*	*d*	

It can be easily seen from the table that this model is realizable monotone.

5.2 Partially Ordered State Spaces

We now consider a partial order on the state space and show that there is only implication but not the equivalence between these two monotonicity definitions.

Theorem 2. *In the case of partially ordered state spaces, if the system is realizable monotone, it is also stochastically monotone.*

Proof. By means of Eq. 2 and definition 9, we can rewrite stochastic monotonicity constraints of matrix $\mathbf{P}$ as follows

$$\forall (x,y)|x \preceq y \text{ and } \forall \Gamma, \sum_{z\in\Gamma} \sum_{u|\phi(x,u)=z} \mathbb{P}(U=u) \leq \sum_{z\in\Gamma} \sum_{u|\phi(y,u)=z} \mathbb{P}(U=u)$$

From the realizable monotone definition, we have for each two states x and $y \in \mathcal{X}$:

$$if\ x \preceq y\ then\ \forall u \in [0,1] : x' = \Phi(x,u) \preceq \Phi(y,u) = y'$$

Thus if x' belongs to an increasing set Γ, then y' belongs to this set (definition (6)). The above inequalities are satisfied for all increasing set Γ, thus $\mathbf{P}$ is st-monotone.

The reciprocal of this implication is not true. We will prove it by a counter example: We consider a transition matrix **P3** in a partially ordered state space. $\mathcal{X} = \{a,b,c,d\}$ and $a \preceq b \preceq d$; $a \preceq c \preceq d$.

$$\mathbf{P}3 = \begin{pmatrix} 1/2 & 1/6 & 1/3 & 0 \\ 1/3 & 1/3 & 0 & 1/3 \\ 1/2 & 0 & 1/6 & 1/3 \\ 0 & 1/3 & 1/3 & 1/3 \end{pmatrix}$$

It can be easily verified that $\mathbf{P}3$ is st-monotone. This model can be described by transition function Φ, obtained by the inverse probability distribution function by considering the total order $a \leqslant b \leqslant c \leqslant d$.

	1/6	1/6	1/6	1/6	1/6	1/6
a	a			b	c	
b	a		b		d	
c	a			c	d	
d	b		c		d	

It can be seen from the table that it is not realizable monotone, for instance, we have for $u \in [3/6, 4/6]$ $\Phi(a,u) = b$ is incomparable with $\Phi(c,u) = c$.

We can not find another transition function which makes this system realizable monotone.

Proposition 1. *It is not possible to construct a realizable monotone transition function for the above example.*

Proof. Since $b \preceq d$ and $c \preceq d$, the transitions from states $b,\ c,\ d$ to state d with probability $1/3$ must be associated to the same interval u. Similarly, since $a \preceq b$ and $a \preceq c$ the transitions from states $a,\ c$ to state a with probability $1/2$ must

be associated to the same interval u, the transitions from states a, b to state a with probability $1/2$ must be associated to the same interval u .

So, for states b, and c it remains only an interval of $u_e = 1/3$ to assign. For b the transition which is not associated is to state b, and for c there are two transitions, one is to state a and the other is to state c. Now, we discuss the case of state a, where $a \preceq c$ and $a \preceq b$. For state a, we have an interval of $1/2$ to assign, the transitions which are not associated are to state b and c. If we associate b to the interval u_e, we have a case of non comparability with state $\Phi(c, u_e) = c$. Similarly, if we associate c to the interval u_e, we will have a case of non comparability with state $\Phi(b, u_e) = b$. Thus it is not possible to build a realizable monotone transition function.

5.3 Case of Equivalence in Partial Order

We will give a case of partial order for which there is an equivalence between the stochastic monotonicity and the realizable monotonicity, we will then give an algorithm to construct the monotone transition function Φ which can be used in PSI 2 to do monotone perfect simulation.

Theorem 3. *When the state space is partially ordered in a tree, if the system is stochastic monotone, then there exists a finite set of events $e_1, e_2, ..., e_n$, for which the system is event-monotone.*

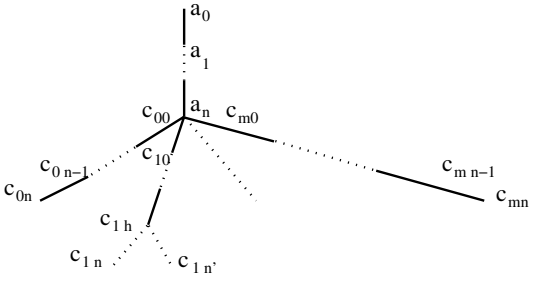


Fig. 2. Tree

We consider one strongly connected component. Let $A = \{a_1 \leq a_2 \leq ...a_n\}$ be the states which are comparable with all others. This set contains at least the root of the tree. $F = \{f_1, ..., f_m\}$ denotes the set of leaves. Suppose that there are m branches from a_n to each leaf f_i. The branches from a_n to f_i are called $C_i = \{c_{i0}, ..., c_{in} = f_i\}$, where c_{i0} is the successor of a_n. Obviously, the states in a branch are totally ordered. We consider branch by branch. and for a given branch C_i, we determine for all states $x \in \mathcal{X}$, events e_h, such that $\Phi(x, e_h) = c_{ij}$. Let N be the number of states in $\mathcal{X}$.

Now we will give the algorithm that construct the monotone transition function Φ, the idea of this algorithm can be summarized as follows: we consider

branch by branch and for branch C_i we find events which trigger transition to a state of C_i. Then we consider the states of A and find events which trigger transition to a state of A.

Algorithm 3. Stochastic monotonicity $\rightarrow$ event-monotonicity

$E = \emptyset$ {the set of events is initially empty}
for $k \in \{1, \cdots m\}$ **do**
  {Consider branch C_k}
  $V = [v_1, v_2, ..., v_N]$ {a vector representing the column index of the rightmost positive values for each row}
  repeat
    for $i \in \{1, 2 \cdots N\}$ **do**
      for $j \in \{v_i, \cdots, c_{k,l}, c_{k,l-1}, \cdots c_{k,0}, a_n, \cdots a_0\}$ **do**
        if $p_{i,j} = 0$ **then**
          $j \leftarrow j + 1$
        end if
      end for
      $v_i \leftarrow j$ {update vector V}
    end for
    $h \leftarrow h + 1$ {the next event e_h}
    $p_{e_h} \leftarrow min_{1 \leq i \leq N}\ p_{i,v_i}$ {probability for event e_h}
    for $i = 1$ to N **do**
      $\Phi(i, e_h) \leftarrow v_i$
      $p_{i,v_i} \leftarrow p_{i,v_i} - p_{e_h}$ {update matrix P}
    end for
  until $\sum_{e_h \in E} p_{e_h} = max_{x \in F}(\sum_{i=0}^{n} p_{x,c_{ki}})$
end for
repeat
  for $i = 1$ to N **do**
    $j \leftarrow v_i$
    while $p_{i,j} = 0$ **do**
      $j \leftarrow j - 1$
    end while
    $v_i \leftarrow j$ {update vector V}
  end for
  $h \leftarrow h + 1$ {the next event e_h}
  $p_{e_h} \leftarrow min_{1 \leq i \leq N}\ p_{i,v_i}$ {probability for event e_h}
  for $i = 1$ to N **do**
    $\Phi(i, e_h) \leftarrow v_i$
    $p_{i,v_i} \leftarrow p_{i,v_i} - p_{e_h}$ {update matrix P}
  end for
until $\sum_{i=1}^{h} p_{e_h} = 1$

Proof. To prove this algorithm, we must show that for all two comparable states x and y, if $x \preceq y$, than we can find a transition function Φ, such that $\Phi(x, u) \preceq \Phi(y, u)$, $\forall u$. For each branch k of the tree, we have from the stochastic monotonicity the following proprieties

$$p_{xc_{kn}} \leq p_{yc_{kn}}$$

$$p_{xc_{kn-1}} + p_{xc_{kn}} \leq p_{yc_{kn-1}} + p_{yc_{kn}}$$

...

$$p_{xc_{k0}} + ... + p_{xc_{kn-1}} + p_{xc_{kn}} \leq p_{yc_{k0}} + ... + p_{yc_{kn-1}} + p_{yc_{kn}}$$

These proprieties satisfy the same conditions of the stochastic monotonicity in a total order. This means that for each branch of the tree, we can construct a monotone transition function by the same method used in the total order. Now, if $\sum_{i=0}^{n} p_{yc_{ki}} > \sum_{i=0}^{n} p_{xc_{ki}}$ we must prove that for all u in the interval, which represent $\sum_{i=0}^{n} p_{yc_{ki}} - \sum_{i=0}^{n} p_{xc_{ki}}$, $\Phi(y,u) \succeq \Phi(x,u)$.

Let

$$diff_k = \sum_{i=0}^{n} p_{yc_{ki}} - \sum_{i=0}^{n} p_{xc_{ki}}$$

So, we must show that the sum of all the differences $diff_k$ is smaller than $(\sum p_{xa} + p_{xb})$. This can be verified in the following equation:

$$\sum p_{xa} + p_{xb} = \sum_{k=0}^{m} diff_k + \sum p_{ya} + p_{yb} \tag{10}$$

$$(10) \Longrightarrow \sum p_{xa} + p_{xb} = \sum_{k=0}^{m}(\sum_{i=0}^{n} p_{yc_{ki}} - \sum_{i=0}^{n} p_{xc_{ki}}) + \sum p_{ya} + p_{yb}$$

$$(10) \Longrightarrow \sum p_{xa} + p_{xb} + \sum_{k=0}^{m}(\sum_{i=0}^{n} p_{xc_{ki}}) = \sum_{k=0}^{m}(\sum_{i=0}^{n} p_{yc_{ki}}) + \sum p_{ya} + p_{yb}$$

$$(10) \Longrightarrow \sum p_{xa} + p_{xb} + \sum_{k=0}^{m}(\sum_{i=0}^{n} p_{xc_{ki}}) = 1 = \sum_{k=0}^{m}(\sum_{i=0}^{n} p_{yc_{ki}}) + \sum p_{ya} + p_{yb}$$

This last equation is evident because of the stochastic proprieties of the matrix.

6 Conclusion

In this paper, we study different monotonicity notions used in stochastic modelling. The stochastic monotonicity associated to stochastic ordering relation and the event and realizable monotonicity is used in perfect simulation. The monotonicity concept depends on the relation order that we consider on the state space. First, we show that if we have a monotone model on a total order, this does not imply that it is monotone in the partial order for both monotonicity notions.

Additionally, we have discussed the relationships between the stochastic monotonicity and the monotonicity used to perform perfect simulation, in order to see whether it is feasible to do monotone perfect simulation on a stochastic monotone models. There are different mathematical tools to build bounding models for complex discrete event systems. In conclusion, under a total order, the

different monotonicity definitions are equivalent to each other. However, under a partial order, the realizable monotonicity implies the stochastic monotonicity. In fact, we have shown that stochastic monotonicity are not sufficient to obtain an event monotone model, but we must verify others conditions on the DTMC. For instance if the partial order is a tree, we have proved that there is an equivalence between the two notions of monotonicity, and we have developed an algorithm which construct the realizable monotone transition function Φ, to do perfect monotone simulation.

References

1. Bolch, G., Greiner, S., de Meer, H., Trivedi, K.: Queueing Networks and Markov Chains. John Wiley & Sons, Chichester (1998)
2. Borovkov, A.A., Foss, S.: Two ergodicity criteria for stochastically recursive sequences. Acta Appl. Math. 34 (1994)
3. Diaconis, P., Freedman, D.: Iterated random functions. SIAM Review 41(1), 45–76 (1999)
4. Fill, J.A., Machida, M.: An interruptible algorithm for perfect sampling via markov chains. In: STOC 1997: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, New York, USA, pp. 688–695 (1997)
5. Fill, J.A., Machida, M.: Realizable monotonicity and inverse probability transform. Technical report, Department of Mathematical sciences. The Johns Hopkins university (2000)
6. Fourneau, J.M., Kadi, I., Pekergin, N., Vienne, J., Vincent, J.M.: Perfect simulation and monotone stochastic bounds. In: ValueTools 2007: Proceedings of the 2nd international conference on Performance 249-263evaluation methodologies and tools, pp. 1–10, ICST (2007)
7. Fourneau, J.M., Pekergin, N.: An algorithmic approach to stochastic bounds. In: Calzarossa, M.C., Tucci, S. (eds.) Performance 2002. LNCS, vol. 2459, pp. 64–88. Springer, Heidelberg (2002)
8. Glasserman, P., Yao, D.: Monotone Structure in Discrete-Event Systems. John Wiley & Sons, Chichester (1994)
9. Olle Haggstrom. Finite Markov Chains and algorithmic applications, Matematisk Statistik, Chalmers teknisk hogshola och Goteborgs universitet (2001)
10. Massey, W.A.: Stochastic ordering for markov processes on partially ordered spaces. Math. Oper. Res. 12(2), 350–367 (1987)
11. Vincent, J.-M., Fernandes, P., Webber, T.: Perfect simulation of stochastic automata networks. In: Al-Begain, K., Heindl, A., Telek, M. (eds.) ASMTA 2008. LNCS, vol. 5055, pp. 249–263. Springer, Heidelberg (2008)
12. Propp, J.G., Wilson, D.B.: Exact sampling with coupled Markov chains and applications to statistical mechanics. Random Structures and Algorithms 9(1&2), 223–252 (1996)
13. Stenflo, O.: Ergodic theorems fory Iterated Function Systems controlled by stochastic sequences. Doctoral thesis n. 14, Umea university (1998)
14. Stoyan, D.: Comparison Methods for Queues and Other Stochastic Models. John Wiley & Sons, Chichester
15. Vincent, J.-M.: Perfect simulation of queueing networks with blocking and rejection. In: SAINT-W 2005: Proceedings of the 2005 Symposium on Applications and the Internet Workshops, Trento, Italy, pp. 268–271 (2005)

Preliminary Results on a Simple Approach to G/G/c-Like Queues

Alexandre Brandwajn[1] and Thomas Begin[2]

[1] Baskin School of Engineering, University of California Santa Cruz, USA
[2] Université Pierre et Marie Curie, LIP6, France
alexb@soe.ucsc.edu, thomas.begin@lip6.fr

Abstract. In this paper we consider a multi-server queue with a near general arrival process (represented as an arbitrary state-dependent Coxian distribution), a near general state-dependent Coxian service time distribution and a possibly finite queueing room. In addition to the dependence on the current number of customers in the system, the rate of arrivals and the progress of the service may depend on each other. We propose a semi-numerical method based on the use of conditional probabilities to compute the steady-state queue length distribution in such a queueing system. Our approach is conceptually simple, easy to implement and can be applied to both infinite and finite $C_m/C_k/c$-like queues. The proposed method uses a simple fixed-point iteration. In the case of infinite queues, it avoids the need for arbitrary truncation through the use of asymptotic conditional probabilities.

This preliminary study examines the computational behavior of the proposed method with a Cox-2 service distribution. Our results indicate that it is robust and performs well even when the number of servers and the coefficient of variation of the service times are relatively high. The number of iterations to attain convergence varies from low tens to several thousand. For example, we are able to solve queues with 1024 servers and the coefficients of variation of the service time and of the time between arrivals set to 4 within 1100 iterations.

Keywords: Multi-server queue, general arrivals, general service times, steady-state queue length distribution, simple efficient semi-numerical solution.

1 Introduction

The use of multiple processing elements or servers to provide an overall high processing capacity is a frequently applied technique in many areas including multi-core processors, distributed systems, storage processors with multiple internal "engines", virtualization in operating systems, where multiple logical CPUs are defined, as well as the Internet where most popular Web sites use multiple"mirror" sites. The numbers of servers in these applications can readily exceed 16 and appears to be growing. Due to intrinsic characteristics of the service demands or the way service is provided, it is possible for the service times and/or inter-arrival times to exhibit high variability. In particular, modern

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 159–173, 2009.

CPUs, storage processors, as well as Web sites make extensive use of internal caches to reduce the expected service time for most requests. The mixture of cache hits and much less frequent cache misses naturally leads to service time distributions characterized by high variability. The potentially high variability of service times is not limited to computer applications [32].

At a high level, the applications described can be viewed as instances of the $G/G/c$ queueing system with a possibly high coefficient of variation of the service time, as well as of the time between arrivals. In real life, the maximum queue depth or buffer capacity is finite. Additionally, in many systems, the rate of service may depend on the current number of customers in the system, e.g. if system overheads increase as the number of customers increases in computer applications. State-dependent arrival rate allows us to represent, for instance, a queue subject to requests generated by a finite set of memoryless sources. In load balancing applications, it is also possible to have arrivals of requests that depend on the progress of service.

We consider a $G/G/c$-like system in which the distribution of the times between arrivals is represented by a Coxian [10] series of memoryless stages. The parameters of this Coxian distribution may depend on the number of customers in the system. The service times are represented by a Coxian distribution generalized to include state-dependent service rates and routing probabilities. Additionally, the rate of arrivals and the progress of the service may depend on each other. A number of authors have studied algorithms for matching an arbitrary distribution by a Coxian, e.g. [7, 25, 12, 19].

We base our method on conditional probabilities, which allows us to derive a computationally efficient semi-numerical approach to the evaluation of the steady-state queue length distribution. The proposed approach, applicable to both finite and infinite $C_m/C_k/c$-like queues, does not rely explicitly on matrix-geometric techniques [20, 22]. It is conceptually simple and appears numerically stable in practice even for large numbers of servers. Unlike certain other approaches (e.g. [23]), our method requires minimal mathematical sophistication and is easy to implement in a standard programming language, which should make it of interest to every-day performance analysts. Results obtained from our method have been verified using discrete-event simulation.

As it is well known (e.g. [30]), in the case of an infinite, state-independent $G/G/c$-like queue, the form of the queue length distribution is asymptotically geometric. Our method exploits this fact to avoid arbitrary truncation present in other methods [29, 27, 22]. For the $C_m/C_k/c$ queue, the coefficient of the asymptotic geometric distribution can be independently obtained from a simple set of equations.

In this paper we present preliminary experimental results on the computational behavior of the proposed approach in the particular case when the service time distribution comprises two stages (generalized Cox-2). It is well known that a standard state-independent two-stage Coxian can be used to match the first two moments of any distribution whose coefficient of variation is greater than

$1/\sqrt{2}$, and a Coxian distribution with an unlimited number of stages (used for the inter-arrival times) can approximate arbitrarily closely any distribution [1].

There is a large body of literature devoted to queues with multiple servers. The computation of the stationary queue length distribution of the $M/M/c$ or the $M/M/c/K$ queue is easy and well known [1]. However, no simple derivation seems to exist, even for the first moment of the queue length, when the service times are not exponentially distributed. For the $M/D/c$ queue, Saaty [24] presents a method to obtain the queue length distribution in steady state, and Cosmetatos [9] proposes an approximate formula to compute the mean waiting time in such a queue. Shapiro [28] considers the $M/E_2/c$ queue, and uses an original state description that leads to a set of differential equations for which he proposes a general solution framework. Mayhugh and McCormick [18], and Heffer [13] expand Shapiro's approach to the $M/E_k/c$ queue for arbitrary values of k. As pointed out by Tijms et al. [31], the solution of the resulting set of differential equations quickly becomes intractable as the value of k or the number of servers increases. Thus, Tijms et al. [31] propose an algorithm to approximately compute the steady-state queue length distribution for the $M/G/c$ queue with variation coefficients up to 3. Hokstad [15] attempts to use the method of supplementary variables to study the $M/K_2/c$ queue, and is able to obtain partial results for up to three servers. The method of supplementary variables is also used by Hokstad [14] and Cohen [8] to derive the stationary queue length distribution for the $M/G/2$ queue. Extensions of this method to a higher number of servers do not appear practical.

Results are even more difficult to obtain when the interarrival time distribution is also general. Ishikawa [16] uses the method of supplementary variables to derive the solution for the $G/E_3/3$ queue. De Smit [11] studies the $G/H/c$ queue but is not able to prove the existence of a solution in the general case, and reports experimental results limited to the $G/H_2/c$ queue. Ramaswami and Lucantoni [21] use the embedded Markov chain approach under the assumption of a phase-type distribution of service times. Their method requires the solution of a non-linear matrix equation. The high order of the matrices involved makes the solution impractical for a higher number of servers. Bertsimas [3] considers the $C_k/C_m/c$ queue and proposes a general method to solve the resulting infinite system of partial differential equations using generating functions. Asmussen and Moller [2] propose a technique to evaluate the distribution of the waiting time in a multi-server queue with phase-type service distributions. The latter two techniques do not appear easy to implement in practice. Several authors consider purely numerical approaches. Takahashi and Takami [29] and Seelen [27] present numerical methods for the $Ph/Ph/c$ queue. Their approach involves an iterative solution of the balance equations using successive aggregation/disaggregation steps. Seelen improves on the initial method proposed by Takashi and Takami by introducing an over-relaxation parameter to speed up convergence. As is often the case, the optimal value of this parameter is not known in advance and a poor choice may interfere with the convergence of the method. Additionally, both methods [29, 27] require arbitrary truncation for a queue with unlimited

queueing room, which can introduce errors. Seelen et al. [26] provide a large number of numerical studies with many different distributions for both the interarrival and service times, the number of servers not exceeding 50. Rhee and Pearce [23] cast this type of queues as a quasi birth and death process [17], and propose a solution but provide no data on its numerical behavior. Some of the approaches proposed in the past even for simpler problems turned out to exhibit computational stability issues (e.g. [35]).

In the next section we describe the queue under study, and we outline our computational approach. We consider first the general case of state-dependent arrival and service rates. We also consider the specific case of an infinite queue where the arrival and service become independent of the number of customers as the latter increases, and the asymptotic queue length distribution for such a system. Section 3 is devoted to numerical results that illustrate the behavior of our method. Although we do not have a theoretical proof of convergence or numerical stability, our preliminary results indicate that the proposed method is computationally stable even with large numbers of servers. Section 4 concludes this paper.

2 Model and Its Solution

We consider the queueing system shown in Figure 1. We denote by n the current number of customers and by c the number of servers in this system. The times between arrivals of customers are represented by a series of m memoryless stages. We use the index j ($j = 1, \ldots, m$) to refer to the current stage of the arrival process. The c servers are assumed to be homogeneous and the service times are represented as a Coxian-like distribution with k memoryless stages. We use the index i ($i = 1, \ldots, k$) to refer to the current stage of the service process when there are customers in the system.. We describe the state of this system in steady state by the triple $(j, \overrightarrow{l}, n)$ where j ($j = 1, \ldots, m$) is the current stage of the arrival process, $\overrightarrow{l} = (l_1, \ldots, l_k)$ is the vector giving the numbers of customers in stages 1 through k of their service, and n is the current number of customers in the system. Note that n refers to customers having completed the arrival process but not yet departed from the system. Note also that it is sufficient to consider stages 2 through k in the vector $\overrightarrow{l}$ since we have $\sum_{i=1}^{k} l_i = \min(n, c)$.

For the service time distribution, the completion rates of the stages and the probability of exiting after each expect the last stage may depend on the current number of users in the system as well as on the current stage of the arrival process. We denote by $\mu_i(n, j)$ the service rate of stage i ($i = 1, \ldots, k$) and by $q_i(n, j)$ the probability that the customer completes its service following stage i when there are n customers in the system and the arrival process is in stage j. We let $\hat{q}_i(n, j) = 1 - q_i(n, j)$ denote the probability that the customer proceeds to stage $i + 1$ upon completion of stage i. We assume that $\mu_i(n, j) > 0$ for $i = 1, \ldots, k$, $0 < \hat{q}_i(n, j) \leq 1$ for $i = 1, \ldots, k - 1$ and $\hat{q}_k(n, j) = 0$. For the interarrival time distribution, we denote by $\lambda_j(n, \overrightarrow{l})$ the completion rate of stage j ($j = 1, \ldots, m$) when the current number of customers in the system is n, the state of the servers is given by $\overrightarrow{l}$, and by $p_j(n, \overrightarrow{l})$ the probability to

complete the arrival process following stage j. $\hat{p}_j(n, \overrightarrow{l}) = 1 - p_j(n, \overrightarrow{l})$ denotes the probability that the customer arrival process proceeds to stage $j+1$ upon completion of the preceding stage. We have $0 < \hat{p}_j(n, \overrightarrow{l}) \leq 1$ for $j = 1, \ldots, m-1$, and $\hat{p}_m(n, \overrightarrow{l}) = 0$ for all values of n. Note that, in the case when there is no state dependency, the arrival process considered can be viewed as simply a renewal process with a Coxian interrenewal distribution. Note also that the stages described may correspond to actual stages of processing and service, or may be just a device to represent non-exponential distributions.

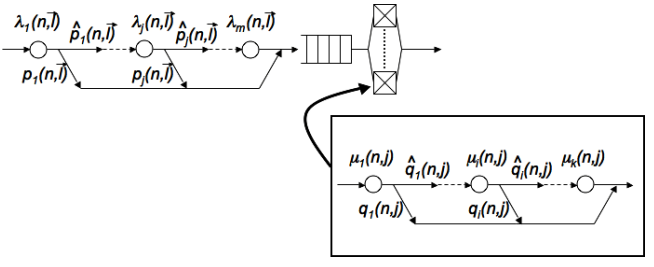


Fig. 1. $C_m/C_k/c$-like queue considered

We let $p(j, \overrightarrow{l}, n)$ be the stationary probability that the system is in the state described by $(j, \overrightarrow{l}, n)$. Denote by $p(j, \overrightarrow{l}|n)$ the corresponding conditional probability that the stage of arrival process is j and that the state of the servers is described by $\overrightarrow{l}$ given that the current number of customers in the system is n. Denote also by $p(n)$ the steady-state probability that there are n customers in the system. Clearly, assuming that $p(n) > 0$, we must have

$$p(j, \overrightarrow{l}, n) = p(j, \overrightarrow{l}|n)p(n). \tag{1}$$

For each value of n, we must also have

$$\sum_{j=1}^{m} \sum_{\overrightarrow{l}} p(j, \overrightarrow{l}|n) = 1. \tag{2}$$

It is a straightforward matter to derive the balance equations for the probabilities $p(j, \overrightarrow{l}, n)$ both in the case of a finite and infinite queueing room. It is not difficult to see that the rate of customer arrivals given n can be expressed as

$$\alpha(n) = \sum_{j=1}^{m} \sum_{\overrightarrow{l}} p(j, \overrightarrow{l}|n)\lambda_j(n, \overrightarrow{l})p_j(n, \overrightarrow{l}), \text{ for } n \geq 0. \tag{3}$$

Similarly, the rate of service completion given that there are n customers in the system can be expressed as

$$\nu(n) = \sum_{j=1}^{m} \sum_{\overrightarrow{l}} p(j, \overrightarrow{l}|n) \sum_{i=1}^{k} l_i \mu_i(n, j) q_i(n, j), \text{ for } n > 0. \tag{4}$$

Hence, $p(n)$, the steady-state probability that there are n customers in the system, is given by

$$p(n) = \frac{1}{G} \prod_{k=1}^{n} \alpha(k-1)/\nu(k), \text{ for } n \geq 0. \tag{5}$$

In formula (5), G is a normalizing constant chosen so that $\sum_{n\geq 0} p(n) = 1$. In other words, the probability $p(n)$ in our $C_m/C_k/c$-like system is the same as the steady-state probability of the number of customers in a simple birth and death process with birth (arrival) rate $\alpha(n)$ and death (service) rate $\nu(n)$. This result can be derived by summing the balance equations for the steady-state probabilities $p(j, \overrightarrow{l}, n)$ over all values of j and $\overrightarrow{l}$, and using the fact that $p(j, \overrightarrow{l}, n) = p(j, \overrightarrow{l}|n)p(n)$ [cf. [6]]. Thus we have (implicit in formula (5))

$$\begin{aligned} p(n-1)/p(n) &= \nu(n)/\alpha(n-1) \\ p(n+1)/p(n) &= \alpha(n)/\nu(n+1). \end{aligned} \tag{6}$$

To obtain the equations for these conditional probabilities $p(j, \overrightarrow{l}|n)$, it suffices to use formula (1) together with (6) in the balance equations for $p(j, \overrightarrow{l}, n)$. In the case of a finite queueing room of size N, there are several possible assumptions regarding the behavior of the arrival process at the high limit resulting in special boundary equations for $n = N$ (and possibly $n = N-1$).

We now focus on the case of an unrestricted queueing room. One possible approach is to simply truncate the equations at some arbitrary high value for n. A more elegant approach is possible if the parameters of the arrival process $\lambda_j(n, \overrightarrow{l})$, $p_j(n, \overrightarrow{l})$, as well as those of the service process $\mu_i(n, j)$, $q_i(n, j)$ become independent of the number of users starting with some value of $n = n_0$ so that we have $\lambda_j(n, \overrightarrow{l}) = \widetilde{\lambda}_j(\overrightarrow{l})$, $p_j(n, \overrightarrow{l}) = \widetilde{p}_j(\overrightarrow{l})$, $\mu_i(n, j) = \widetilde{\mu}_i(j)$, $q_i(n, j) = \widetilde{q}_i(j)$ for $n \geq n_0$. Under these conditions, and assuming that the system under consideration is ergodic, one can expect that the conditional probabilities $p(j, \overrightarrow{l}|n)$ tend to a limit as n increases: $\lim_{n\to\infty} p(j, \overrightarrow{l}|n) = \widetilde{p}(j, \overrightarrow{l})$.

As a result, starting with a sufficiently high value of n, say $n \geq \widetilde{n}$ (clearly, $\widetilde{n} > n$), we have for $||p(j, \overrightarrow{l}|n) - \widetilde{p}(j, \overrightarrow{l})|| < \delta$ for $\delta > 0$, and the arrival and departure rates $\alpha(n)$ and $\nu(n)$ become sufficiently close to their limiting values, which we denote by $\widetilde{\alpha}$ and $\widetilde{\beta}$

$$\widetilde{\alpha} = \sum_{j=1}^{m} \sum_{\overrightarrow{l}: l_1+\ldots+l_k=c} \widetilde{p}(j, \overrightarrow{l})\widetilde{\lambda}_j(\overrightarrow{l})\widetilde{p}_j(\overrightarrow{l}) \tag{7}$$

$$\widetilde{\nu} = \sum_{j=1}^{m} \sum_{\overrightarrow{l}: l_1+\ldots+l_k=c} \widetilde{p}(j, \overrightarrow{l}) \sum_{i=1}^{k} l_i \widetilde{\mu}_i(j)\widetilde{q}_i(j). \tag{8}$$

Thus, we can express the steady-state distribution $p(n)$ as

$$p(n) \approx \frac{1}{G} \begin{cases} \prod_{k=1}^{n} \alpha(k-1)/\nu(k), & n \leq \widetilde{n} \\ \prod_{k=1}^{\widetilde{n}} \alpha(k-1)/\nu(k)(\widetilde{\alpha}/\widetilde{\nu})^{n-\widetilde{n}}, & n > \widetilde{n} \end{cases} \tag{9}$$

Following a common convention, empty products are set to one. The normalizing constant G can be written as

$$G \approx 1 + \sum_{n=1}^{\widetilde{n}-1} \prod_{k=1}^{n} \alpha(k-1)/\nu(k) + \left[\prod_{k=1}^{\widetilde{n}} \alpha(k-1)/\nu(k) \right] \frac{1}{1-(\widetilde{\alpha}/\widetilde{\nu})}, \tag{10}$$

and the expected number of customers in the system can be expressed as

$$\overline{n} \approx \frac{1}{G} \left\{ \sum_{n=1}^{\widetilde{n}} np(n) + \left[\frac{\widetilde{n}}{1-(\widetilde{\alpha}/\widetilde{\nu})} + \frac{(\widetilde{\alpha}/\widetilde{\nu})}{[1-(\widetilde{\alpha}/\widetilde{\nu})]^2} \right] \prod_{k=1}^{\widetilde{n}} \alpha(k-1)/\nu(k) \right\}. \tag{11}$$

We note that the form of the solution for $p(n)$ given in formula (9) clearly shows that the steady-state distribution is asymptotically geometric with "traffic intensity" $\widetilde{\alpha}/\widetilde{\nu}$.

Thus, we solve the set of equations for the conditional probabilities $p(j, \overrightarrow{l}\,|n)$ for all values of n, subject to the normalizing condition given by (2). In the case of an infinite queue, the values to consider are $n = 0, \ldots, \widetilde{n}$, and in the case of a finite queueing room, all values of $n = 0, \ldots, N$. Because the equations for $p(j, \overrightarrow{l}\,|n)$ involve in general the conditionals for $n-1$ and $n+1$, it does not seem possible to solve these equations as a simple recurrence, as would be the case for an $M/G/1$-like queue (cf. [5]).

However, a simple-minded and simple to implement fixed-point iteration can be used to solve these equations as follows. We use a superscript to denote the iteration number. We start with some set of initial values $p^0(j, \overrightarrow{l}\,|n)$ for $n = 0, \ldots, n_{max}$ (where $n_{max} = N$ in the case of a finite queueing room, and $n_{max} = \widetilde{n}^0$, an initial estimate of $\widetilde{n}$, in the case of an infinite queue), and we consider the possible states in the order of increasing n, enumerating all server states $\overrightarrow{l}$ compatible with the value of n, and j, the latter varying the fastest. We compute new values for the conditional probabilities directly from the corresponding equations. For each value of n, we normalize the newly computed values so that $\sum_{j=1}^{m} \sum_{\overrightarrow{l}} p^i(j, \overrightarrow{l}\,|n) = 1$ once we have updated all the values for all $(j, \overrightarrow{l}\,)$, but in the iteration we use the latest (not necessarily normalized) values as soon as they become available. Following the normalization, we can compute new values for the conditional rate of request arrivals $\alpha^i(n)$ and the rate of completions $\nu^i(n)$.

In the case of an infinite queue (under the assumptions discussed earlier in this section) we dynamically determine the "cutoff" value $\widetilde{n}^i$ as the value of n for which $|1 - \alpha^i(n-1)/\alpha^i(n)| < \epsilon$, as well as $|1 - \nu^i(n-1)/\nu^i(n)| < \epsilon$, and we consider that at this point the limiting values have been reached for

the conditional probabilities at iteration i. Note that by selecting the value of ϵ as desired we control the accuracy with which the convergence to limiting conditional probabilities is determined. Thus our method provides an automatic limitation for the values of n based on the accuracy of convergence to limiting values, as opposed to arbitrary truncation used in several other methods. In practice, in most cases, the convergence to limiting values tends to occur quickly (i.e., for moderate values of $\widetilde{n}^i$), so that the steady-state distribution can be determined with high accuracy at a limited computational expense. For a finite queueing room, the maximum value for n is the size of the queueing room N, and there is no asymptotic convergence involved. The fixed-point iteration itself stops when the values of the conditional probabilities at consecutive iterations differ less than a specified convergence tolerance, e.g.$||1-p^{i-1}(j, \overrightarrow{l}|n)/p^i(j, \overrightarrow{l}|n)|| < \delta$.

The fact that we use newly computed values for $p^i(j, \overrightarrow{l}|n)$ as soon as they become available not only reduces the space requirements of our method to a single set of arrays to hold the values of $p^i(j, \overrightarrow{l}|n)$, $\alpha^i(n)$ and $\nu^i(n)$, but also appears to speed up the convergence. We have not been successful in developing a theoretical proof of convergence for the proposed approach.

In our initial study of the properties of this approach, we performed a large number of test runs concentrated on the particular case of a Cox-2 service distribution, for which $p(j, \overrightarrow{l}|n)$ can be replaced by $p(j, l_2|n)$. In our test runs, the proposed approach has always converged, typically within a relatively small number of iterations although the number of iterations tends to increase as the number of servers and the service time variability increase. The choice of the initial distribution $p^0(j, l_2|n)$ seems to have a limited effect. For each value of n, the computational complexity of every iteration scales linearly with the number of servers since the latter determines the number of values to consider for the current number of customers in their second stage of service, l_2. As discussed in the next section, for the unrestricted queue, the value of $\widetilde{n}$, and hence the number of values of n to consider, appears to increase less than linearly as the number of servers increases. Obviously, in the case of a finite queueing room, we have $n = 0, \ldots, N$ at each iteration.

In the case of an infinite queueing room, it is possible to know independently, from the solution of the corresponding equations, the limiting distribution $\widetilde{p}(j, l_2)$. We find that using this limiting distribution for $p^0(j, l_2|n)$ (truncated and normalized for $n < c$), tends to speed up the iteration. It can be readily computed from the equations for $\widetilde{p}(j, l_2)$ using a simple fixed-point iteration.

In the next section, we present numerical results to illustrate the behavior of our method for a number of values of queue parameters, including service times with high variability (coefficient of variation of over 10) and number of servers c ranging from 4 to 256.

3 Numerical Results

In this section we present numerical results to illustrate the good convergence properties of our method, as well as its ability to solve systems both with high number of servers and high service time variability. In most examples we consider

Table 1. Parameters of selected service time distributions

Dist.	Mean	Coeff.Var.	Skewness	Kurtosis	μ_1	μ_2	$\hat{q}_1$
I	1	0.8	1.80	5.05	4.25	1.308	1.000
II	1	2.0	3.06	12.77	1000.0	0.400	0.399
III	1	4.0	6.01	48.28	1000.0	0.118	0.117
IV	1	8.0	12.01	192.43	1000.0	0.031	0.031
V	1	16.0	24.02	769.55	1000.0	0.008	0.008

three levels of server utilization: 0.25, 0.5, 0.99, which correspond to 25%, 50% and 99% of the c servers busy, respectively. The Cox-2 distributions used to represent the service times in our examples are given in Table 1. Note that skewness and kurtosis relate to moments of order 3 and 4 of a probability distribution. Results in the following figures are then labeled by the corresponding coefficient of variation of the service time distribution. The mean service time is kept at one in all cases. We used discrete-event simulation to confirm the accuracy of our results for a selected set of cases.

With infinite queueing room, the "cutoff" point for the determination of $\widetilde{n}^i$ was obtained using $\epsilon = 10^{-11}$. The overall iteration convergence criterion used was $||1-\nu^{i-1}(n)/\nu^i(n)|| < \delta$ and $||1-\alpha^{i-1}(n)/\alpha^i(n)|| < \delta$ with $\delta = 10^{-5}$. These values were used for all examples presented in this paper.

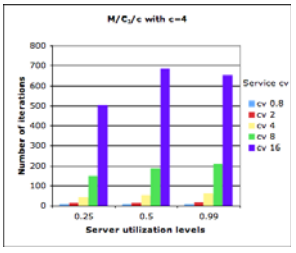
3.1 The $M/G/c$-Like Queue

In our first set of results we consider an infinite state-independent queue with Poisson arrivals, i.e., an $M/C_2/c$ queue. Figures 2a through 2h show the number of iterations needed to achieve convergence as well as the largest values of $\widetilde{n}^i$ observed during the iteration process (thus indicating the number of equations solved and storage requirements.)

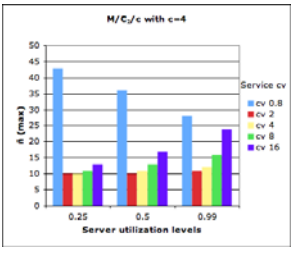
We observe that the number of iterations is generally tame, ranging from no more than around 200 for coefficients of variation up to 8 and 4 servers, to below 1000 with 256 servers. In our examples, the number of iterations tends to increase as the coefficient of variation of the service time increases, although, as we discuss later in this section, the results can be quite sensitive to higher order parameters of the service time distribution. When the coefficient of variation of the service time is equal to 16, the number of iterations ranges from around 700 to below 4000. The convergence of $p(j, l_2|n)$ to the limiting distribution $\widetilde{p}(j, l_2)$ as n increases tends to occur relatively quickly. The maximum values of $\widetilde{n}^i$ attained during the iteration range from low tens to below 1000 in the "worst" case for the queue considered, viz. for 256 servers and coefficient of variation of the service set to 16.

3.2 The $M/G/c/N/N$-Like Queue

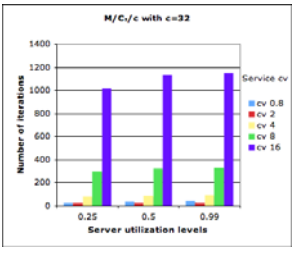
In our second set of results we consider a similar queue subject to state dependent memoryless arrivals, i.e., the rate of arrivals when there are n requests in the queue (including the ones in service) is given by $\lambda(n) = (N-n)\gamma$. Such a model corresponds to a set of N sources of requests as shown in Figure 3. Each source generates a new request after an exponentially distributed time $1/\gamma$ following the completion of its previous service period.



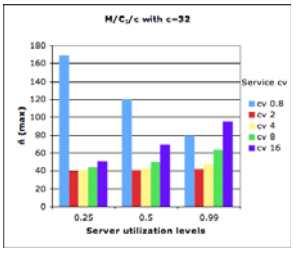
(a) Number of iterations with 4 servers.



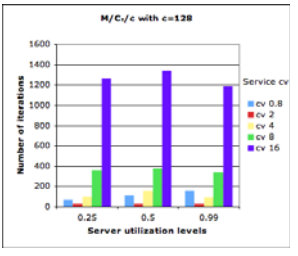
(b) Maximum value of $\widetilde{n}^i$ with 4 servers.



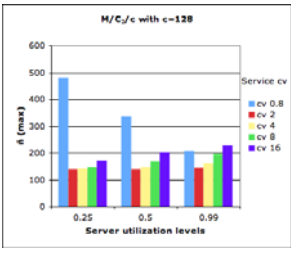
(c) Number of iterations with 32 servers.



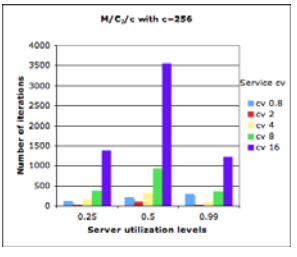
(d) Maximum value of $\widetilde{n}^i$ with 32 servers.



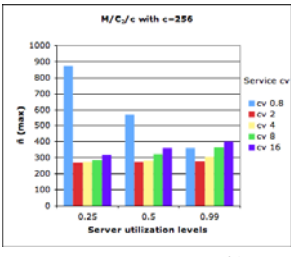
(e) Number of iterations with 128 servers.



(f) Maximum value of $\widetilde{n}^i$ with 128 servers.



(g) Number of iterations with 256 servers.



(h) Maximum value of $\widetilde{n}^i$ with 256 servers.

Fig. 2. Behavior of the method for a multi-server queue for service time distributions from Table 1 as a function of the number of servers c and the server utilization level

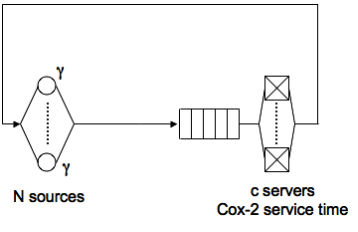
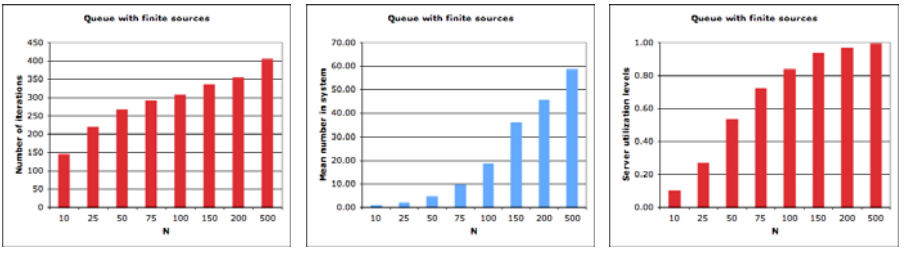


Fig. 3. Multi-server queue with N sources

The results shown in Figure 4 pertain to a queue with 8 servers and a coefficient of variation of the service time of 8. Figures 4a, 4b and 4c show the number of iterations, the expected number of customers in the system (queued and in service) $\bar{n}$, as well as the utilization level (fraction of servers busy), respectively,



(a) Number of iterations (b) Mean number of customers (c) Server utilization levels

Fig. 4. Behavior of the method for a multi-server queue with $c = 8$ servers and service time distribution Dist. IV ($cv = 8$, cf. Table 1) as a function of the number of sources N.

for numbers of sources ranging from 10 to 500. The value of γ is kept at 0.1. We observe that the number of iterations to achieve convergence tends to increase with the number of sources, but remains, in the example considered, below 500 in all cases.

3.3 The $G/G/c$-Like Queue

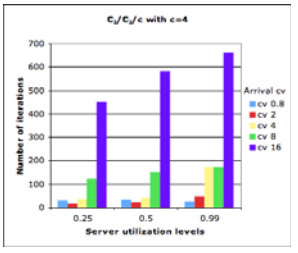
In Figure 5 we have represented results for a $C_2/C_2/c$ queue with infinite queueing room in which the time between consecutive arrivals is a Cox-2 distribution with a coefficient of variation of 4. The parameters of the service time distribution are given in Table 1. The generic parameters of the distributions of the interarrival times used in our examples are given in Table 2. The values given in this table correspond to a mean time between arrivals of one. For other values of the mean interarrival time used in our examples, the rates of the stages of the arrival process change in proportion to the inverse of that mean, while the stage transition probabilities remain constant.

Table 2. Generic parameters of selected distributions of time between arrivals

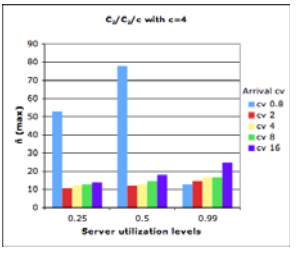
Dist.	Mean	Coeff.Var.	Skewness	Kurtosis	λ_1	λ_2	$\hat{p}_1$
I	1	0.8	1.80	5.05	4.248	1.308	1.000
II	1	2.0	3.36	15.31	10.00	0.375	0.338
III	1	4.0	6.66	59.30	10.00	0.107	0.096
IV	1	8.0	13.33	236.96	10.00	0.028	0.025
V	1	16.0	26.66	948.05	10.00	0.007	0.006

We give in Tables 1 and 2 the precise parameters of the distributions used because the results can be sensitive to higher order parameters of both the interarrival time distribution and the service time distribution [4, 33, 34]. This sensitivity extends to the performance of our method, as well as the steady-state probability distribution for the $G/G/c$ queue itself.

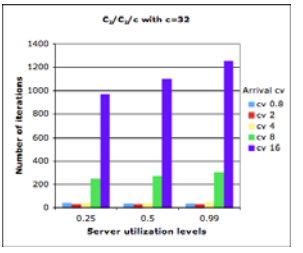
The number of iterations ranges typically from 200 with 4 servers to 500 with 256 servers when the coefficient of variation of the service time does not exceed 8. With the coefficient on variation of the service time set to 16, the number of iterations ranges from about 700 with 4 servers to 3500 with 256 servers. The



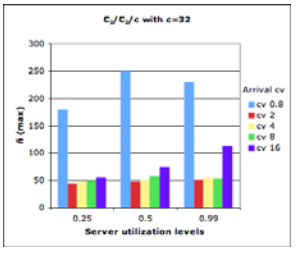
(a) Number of iterations with 4 servers



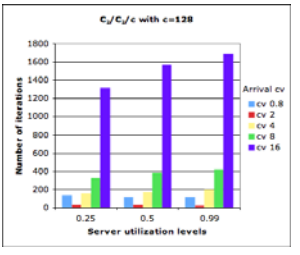
(b) Maximum value of $\widetilde{n}^i$ with 4 servers



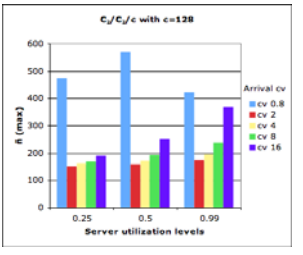
(c) Number of iterations with 32 servers



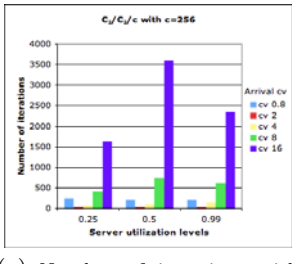
(d) Maximum value of $\widetilde{n}^i$ with 32 servers



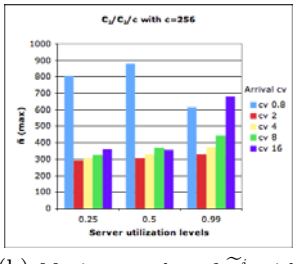
(e) Number of iterations with 128 servers



(f) Maximum value of $\widetilde{n}^i$ with 128 servers



(g) Number of iterations with 256 servers



(h) Maximum value of $\widetilde{n}^i$ with 256 servers

Fig. 5. Behavior of the method for a multi-server queue for inter-arrivals time distributions from Table 2 as a function of the number of servers c and the server utilization level

maximum values of $\widetilde{n}^i$ attained during the iteration range from low tens to below 1000 in the "worst" case, which happens to be in this case for 256 servers and coefficient of variation of the service set to 0.8.

Using a "proof-of-concept" implementation in C running on a 2.99 GHz Intel processor, for the case of 128 servers at 0.99 server utilization level, we measured execution times ranging from 3.97 s with $cv = 8$ to around 0.31 s with a lower coefficient of variation of 2 ($cv = 2$). As mentioned before, lower numbers of servers tend to result in faster execution, so that with 32 servers the corresponding results range from 0.72 s to 0.16 s. With 4 servers the execution times of our simple implementation range from about 0.14 s for $cv = 8$ to 0.03 s for $cv = 2$. Note that the vast majority of execution times are below one second.

The convergence stringency used throughout this paper, viz. $\epsilon = 10^{-11}$ and $\delta = 10^{-5}$ appears generally sufficient. When focusing on individual state probabilities in our trials, we used more stringent values: $\epsilon = 10^{-15}$ and $\delta = 10^{-8}$. There seems to be limited difference in the results obtained.

Overall, our method appears to be computationally robust, reasonably fast and quite scalable as the number of servers and the variability of service and interarrival times increase. The next section is devoted to the conclusions of this paper.

4 Conclusion

We consider a semi-numerical method to compute the steady-state distribution of the number of users in a $C_m/C_k/c$-like system where the distributions of the times between arrivals and the service times are represented by Coxian series of memoryless stages. The parameters of both Coxian distributions may depend on the current number of customers in the system. Additionally, arrivals and the progress of the service may depend on each other. We base our approach explicitly on conditional probabilities. This allows us to derive a conceptually simple and computationally efficient semi-numerical approach to the evaluation of the steady-state queue length distribution.

The proposed method can be used to solve both infinite and finite $G/G/c$-like queues of the type considered. In the case of an infinite $C_m/C_k/c$ queue whose parameters don't depend on the current number of customers, the form of the queue length distribution is asymptotically geometric. Our method exploits this fact to avoid arbitrary truncation of the balance equations. Instead, we dynamically determine, with as much stringency as desired, the convergence to asymptotic values, and use the latter in our solution. The coefficient of the geometric distribution is a by-product of our iterative solution. It can also be obtained independently, without solving the whole queue, using a simple set of equations, easily solved via fixed-point iteration.

In this preliminary study, we examined empirically the computational properties of this method in the case of a Cox-2 service distribution. Our experimental evidence indicates that the proposed method is numerically stable in practice. In our numerical examples we have explored the behavior of our approach for a range of values of the number of servers in the queue (4 to 256), as well as for several coefficients of variation of the time between arrivals and of the service times. Our results indicate that the proposed method performs well even when the number of servers is relatively high (256 in our examples) and so is the coefficient of variation (up to 16 in our examples). In the many cases we considered, the method has never failed to converge within a reasonable number of iterations. The number of iterations to attain convergence depends on the parameters of the $C_m/C_2/c$ queue considered, and varies, in our examples, from low tens to several thousand. It tends to increase for queues with high coefficients of variation of the service time and high number of servers. In additional tests, not reported in Section 3, we were able to solve queues with 1024 servers, the number of iterations not exceeding 1100 for the coefficients of variation of the service time and of the time between arrivals set to 4.

Our results underscore the potential importance of higher order moments of the interarrival time and service time distributions in the steady-state probability distribution for the number of customers in the $G/G/c$ queue. This topic is discussed in more detail in another paper.

Overall, the proposed method is conceptually simple, easy to implement, and readily applicable to both finite and infinite systems. It requires minimal mathematical sophistication. Our preliminary results indicate that it robust, fast, and scales reasonably well with the number of servers. These qualities should make the method attractive to performance analysts "in the trenches" when dealing with systems that can be modeled as mutliserver queues.

References

1. Allen, A.O.: Probability, Statistics, and Queueing Theory with Computer Science Applications, 2nd edn. Academic Press, London (1990)
2. Asmussen, S., Moller, J.R.: Calculation of the Steady State Waiting Time Distribution in $GI/PH/c$ and $MAP/PH/c$ Queues. Queueing Systems 37, 9–29 (2001)
3. Bertsimas, D.: An Analytic Approach to a General Class of $G/G/s$ Queueing Systems. Operations Research 38(1), 139–155 (1990)
4. Bondi, A.B., Whitt, W.: The influence of service-time variability in a closed network of queues. Performance Evaluation 6(3), 219–234 (1986)
5. Brandwajn, A., Wang, H.: A Conditional Probability Approach to $M/G/1$-like Queues. Performance Evaluation 65(5), 366–381 (2008)
6. Brandwajn, A.: Equivalence and Decomposition in Queueing Systems - A Unified Approach. Performance Evaluation 5, 175–185 (1985)
7. Bux, W., Herzog, U.: The Phase Concept: Approximation of Measured Data and Performance Analysis. In: Proceedings of the International Symposium on Computer Performance Modeling, Measurement and Evaluation, Yorktown Heights, NY, pp. 23–38. North-Holland, Amsterdam (1977)
8. Cohen, J.W.: On the the $M/G/2$ Queueing Model. Stochastic Processes and Their Applications 12, 231–248 (1982)
9. Cosmetatos, G.P.: Approximate Explicit Formulae for the Average Queueing Time in the Process $(M/D/r)$ and $(D/M/r)$. INFOR 13, 328–331 (1975)
10. Cox, D.R., Smith, W.L.: Queues. John Wiley, New York (1961)
11. De Smit, J.H.A.: The Queue $GI/M/s$ with Customers of Different Types or the Queue $GI/Hm/s$. Advances in Applied Probability 15(2), 392–419 (1983)
12. Faddy, M.: Penalised Maximum Likelihood Estimation of the Parameters in a Coxian Phase-Type Distribution. In: Matrix-Analytic Methods: Theory and Application: Proceedings of the Fourth International Conference, Adelaide, Australia, pp. 107–114 (2002)
13. Heffer, J.C.: Steady State Solution of the $M/Ek/c$ (infinty, FIFO) queueing system. INFOR. 7, 16–30 (1969)
14. Hokstad, P.: On the Steady-State Solution of the $M/G/2$ Queue. Advances in Applied Probability 11(1), 240–255 (1979)
15. Hokstad, P.: The steady-state solution ok the $M/K2/m$ Queue. Advances in Applied Probability 12(3), 799–823 (1980)
16. Ishikawa, A.: On the equilibrium solution for the Queueing System $GI/Ek/m$. TRU Mathematics 15, 47–66 (1979)

17. Latouche, G., Ramaswami, V.: Introduction to Matrix Analytic Methods in Stochastic Modeling, ASA (1999)
18. Mayhugh, J.O., McCormick, R.E.: Steady State Solution of the Queue $M/Ek/r$. Management Science, Theory Series 14(11), 692–712 (1968)
19. McLean, S., Faddy, M., Millard, P.: Using Markov Models to assess the Performance of a Health and Community Care System. In: Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems, pp. 777–782 (2006)
20. Neuts, M.F.: Matrix-geometric solutions in stochastic models. An algorithmic approach. Courier Dover Publications (1994)
21. Ramaswami, V., Lucantoni, D.M.: Stationary waiting time distribution in queues with phase type service and in quasi-birth-and-death-processes. Stochastic Models 1, 125–136 (1985)
22. Ramaswami, V., Lucantoni, D.M.: Algorithms for the multi-server queue with phase type service. Stochastic Models 1, 393–417 (1985)
23. Rhee, K.H., Pearce, C.E.M.: On Some Basic Properties of the Inhomogeneous Quasi-Birth-And-Death Process. Comm. Korean Math. Soc. 12(1), 177–191 (1997)
24. Saaty, T.L.: Elements of Queueing Theory, with Application. The Annals of Mathematical Statistics 34(4), 1610–1612 (1963)
25. Sasaki, Y., Imai, H., Tsunoyama, M., et al.: Approximation of probability distribution functions by Coxian distribution to evaluate multimedia systems. Systems and Computers in Japan 35(2), 16–24 (2004)
26. Seelen, L.P., Tijms, H.C., Van Hoorn, M.H.: Tables for Multi-Server Queues. North-Holland, Amsterdam (1984)
27. Seelen, L.P.: An Algorithm for $Ph/Ph/c$ Queues. European Journal of the Operations Research Society 23, 118–127 (1986)
28. Shapiro, S.: The M-Server Queue with Poisson Input and Gamma-Distributed Service of Order Two. Operations Research 14(4), 685–694 (1966)
29. Takahashi, Y., Takami, Y.: A Numerical Method for the Steady-State Probabilities of a $GI/G/s$ Queueing system in a General Class. Journal of the Operations Research Society of Japan 19, 147–157 (1976)
30. Takahashi, Y.: Asymptotic Exponentiality of the Tail of the Waiting-Time Distribution in a $PH/PH/c$ Queue. Advances in Applied Probability 13(3), 619–630 (1981)
31. Tijms, H.C., Van Hoorn, M.H., Federgruen, A.: Approximations for the Steady-State Probabilities in the $M/G/c$ Queue. Advances in Applied Probability 13(1), 186–206 (1981)
32. van Dijk, N.M.: Why queuing never vanishes. European Journal of Operational Research 99(2), 463–476 (1997)
33. Whitt, W.: The Effect of Variability in the $GI/G/s$ Queue. Journal of Applied Probability 17(4), 1062–1071 (1980)
34. Wolff, R.W.: The Effect of Service Time Regularity On System Performance. In: Computer Performance, pp. 297–304. North Holland, Amsterdam (1977)
35. Ye, Q.: On Latouche-Ramaswami's logarithmic reduction algorithm for quasi-birth-and-death processes. Comm. Stat. & Stochastic Models 18, 449–467 (2002)

Moments Characterization of Order 3 Matrix Exponential Distributions*

András Horváth[1], Sándor Rácz[2], and Miklós Telek[2]

[1] Dipartimento di Informatica, University of Torino, Italy
[2] Technical University of Budapest, Hungary
horvath@di.unito.it, sandor.racz.74@gmail.com, telek@hit.bme.hu

Abstract. The class of order 3 phase type distributions (PH(3)) is known to be a proper subset of the class of order 3 matrix exponential distributions (ME(3)). In this paper we investigate the relation of these two sets for what concerns their moment bounds. To this end we developed a procedure to check if a matrix exponential function of order 3 defines a ME(3) distribution or not. This procedure is based on the time domain analysis of the density function. The proposed procedure requires the numerical solution of a transcendent equation in some cases.

The presented moment bounds are based on some unproved conjectures which are verified only by numerical investigations.

Keywords: Matrix exponential distributions, Phase type distributions, moment bounds.

1 Introduction

The availability of efficient matrix analytic methods (see e.g., [7,10]) reinforced the research of distributions with matrix exponential representation. The order of these distributions is defined as the (minimal) cardinality of the matrix that describes the distribution. The two main classes of these distributions are the class of phase type distributions [8,9], which has a nice stochastic interpretation due to its underlying continuous time Markov chain, and the class of matrix exponential distributions [1], which does not allow for a simple stochastic interpretation.

It has been known for a long time that considering distributions of order 2 the two classes are identical, ME(2)≡PH(2), but for $n > 2$ PH(n) is a proper subset of ME(n) [12]. Unfortunately there are no tools to investigate the relation of the ME(n) and the PH(n) classes for $n > 2$. However, recent results on ME(3) [5] and PH(3) [6] distributions make it possible to investigate the relation of the ME(3) and the PH(3) classes.

The practical importance of low order PH and ME distributions comes from the fact that the complexity of the matrix analytic analysis increases rapidly

* This work is partially supported by the NAPA-WINE FP7 project and by the OTKA K61709 grant.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 174–188, 2009.

with the order of the model components (e.g., PH distribution of the service time). Recent results suggest that matrix analytic methods are applicable for models with matrix exponential distributions as well as for models with phase type distributions [2]. Consequently, one can gain if the durations to be modelled can be described by a ME distribution with lower order than the application of a PH distributions would require.

We compare the flexibility of the ME(3) and the PH(3) classes through their moment bounds. It is not the only and not necessarily the easiest way to compare them, but this choice is motivated by the fact that moments and related measures (e.g., coefficient of variation) are the most frequently used parameters of distributions.

This paper is strongly related to the extensive work of Mark Fackrell in [5]. We reconsider some questions of [5] and complement those results with alternative ones. The main goal of this paper is to answer the following question ([5] p. 110) "The class of PH distributions is a proper subset of the class of ME distributions, but how much larger is the latter class than the former?". In [5] the question is answered for ME(n) and $\cup_{m \geq n}$ PH(m). We believe that this question has more practical importance for ME(n) and PH(n). In this work we try to answer this question for ME(3) and PH(3).

Related ME(3) results. [5] devotes its main attention to the matrix exponential distributions of order $n > 2$ and provides important necessary conditions for being a member of ME(n). Additionally, [5] provides necessary and sufficient conditions for being a member of ME(3). These conditions are given in the Laplace transform domain. Assuming that for a given triple $\{b_1, b_2, b_3\}$ the Laplace transform of a matrix exponential function takes the form

$$\frac{x_2 s^2 + x_1 s + b_1}{s^3 + b_3 s^2 + b_2 s + b_1}$$

(i.e., there is no probability mass at 0) the linear and parametric curves provided in [5] bound the region of $\{x_1, x_2\}$ where the matrix exponential function is a member of the ME(3) class.

Unfortunately, we did not find an easy implementation of these transform domain constraints, and this is why we developed a time domain counterpart for ME(3) characterization.

An important property of the ME(3) class, namely its minimal coefficient of variation, is studied in [4]. The results provided here verify the ones provided there.

Related PH(3) results. Another important preliminary work is [6] which provides a canonical representation of PH(3) distributions. More precisely, [6] presents an algorithm that transforms any order 3 matrix exponential function to PH(3) canonical form if it is possible. In this paper, this algorithm is used to characterize the borders of the PH(3) class.

The rest of the paper is organized as follows. Section 2 defines the class of matrix exponential distributions and the basic notations. Section 3 presents a procedure to check if a matrix exponential function of order 3 is a member of

the ME(3) class or not. Using this procedure and its counterpart for the PH(3) class from [6], Section 4 investigates the relation of the moment bounds of these two classes. The paper is concluded in Section 5.

2 Matrix Exponential Distributions

Definition 1. *The vector matrix pair* $(\underline{v}, \boldsymbol{H})$ *defines a matrix exponential distribution iff*

$$F(t) = Pr(X < t) = 1 - \underline{v} e^{\boldsymbol{H} t} \mathbb{1} , \qquad t \geq 0 \tag{1}$$

is a valid cumulative distribution function, i.e., $F(0) \geq 0$, $\lim_{t\to\infty} F(t) = 1$ *and* $F(t)$ *is monotone increasing.*

In (1), the row vector, $\underline{v}$, is referred to as the initial vector, the square matrix, $\boldsymbol{H}$, as the generator and $\mathbb{1}$ as the closing vector. Without loss of generality (see [8]), throughout this paper we assume that the closing vector is a column vector of ones, i.e., $\mathbb{1} = [1, 1, \ldots, 1]^T$.

The density, the Laplace transform and the moments of the matrix exponential distribution defined by $(\underline{v}, \boldsymbol{H})$ are

$$f(t) = \underline{v} e^{\boldsymbol{H} t} (-\boldsymbol{H}) \mathbb{1} , \tag{2}$$

$$f^*(s) = E(e^{-sX}) = \underline{v} (s\boldsymbol{I} - \boldsymbol{H})^{-1} (-\boldsymbol{H}) \mathbb{1} , \tag{3}$$

$$\mu_n = E(X^n) = n! \underline{v} (-\boldsymbol{H})^{-n} \mathbb{1} . \tag{4}$$

To ensure that $\lim_{t\to\infty} F(t) = 1$, $\boldsymbol{H}$ has to fulfill the necessary condition that the real parts of its eigenvalues are negative (consequently $\boldsymbol{H}$ is non-singular).

The remaining constraint is the monotonicity of $F(t)$. It is the most difficult property to check. Instead of checking if $F(t)$ is monotone increasing, in the next section, we check if $f(t)$ is non-negative.

3 Matrix Exponential Distributions of Order 3

We subdivide the class of ME(3) distributions according to the eigenvalue structure of $\boldsymbol{H}$. With $\lambda_1, \lambda_2, \lambda_3$ denoting the eigenvalues of the matrix $-\boldsymbol{H}$, we have the following possible cases:

- class A: $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}^+$, $\lambda_1 < \lambda_2 < \lambda_3$
- class B: $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}^+$, $\lambda_1 = \lambda_2 < \lambda_3$ or $\lambda_1 < \lambda_2 = \lambda_3$
- class C: $\lambda_1 = \lambda_2 = \lambda_3 \in \mathbb{R}^+$,
- class D: $\lambda_1 \in \mathbb{R}^+$, $\lambda_2 = \overline{\lambda_3} \in \mathbb{C}^+$,

where $\mathbb{R}^+$ denotes the set of strictly positive real numbers and $\mathbb{C}^+$ the set of complex numbers with strictly positive real part. The following subsections consider these four cases.

3.1 Case A: 3 Different Real Eigenvalues

In this case the general form of the density function and its derivative are

$$f(t) = a_1 e^{-\lambda_1 t} + a_2 e^{-\lambda_2 t} + a_3 e^{-\lambda_3 t} \tag{5}$$

$$f'(t) = -a_1\lambda_1 e^{-\lambda_1 t} - a_2\lambda_2 e^{-\lambda_2 t} - a_3\lambda_3 e^{-\lambda_3 t} \tag{6}$$

Without loss of generality, we check the non-negativity of $f(t)$ assuming that $\lambda_1 < \lambda_2 < \lambda_3$.

Theorem 1. *$f(t)$ is non-negative for $t \geq 0$ iff*

- $a_1 + a_2 + a_3 \geq 0$ *and*
- $a_1 > 0$ *and*
- *if* $a_2 < -a_1 \frac{\lambda_3-\lambda_1}{\lambda_3-\lambda_2}$ *then* $a_3 \geq a_1 \frac{\lambda_2-\lambda_1}{\lambda_3-\lambda_2} \left(-\frac{a_2}{a_1} \frac{\lambda_3-\lambda_2}{\lambda_3-\lambda_1}\right)^{\frac{\lambda_3-\lambda_1}{\lambda_2-\lambda_1}}$.

Proof. First, we note that $f(t)$ is a monotone increasing function of a_1, a_2 and a_3 for $t \geq 0$ and both $f(t)$ and $f'(t)$ can have at most 2 roots in $(0, \infty)$ (excluding 0 and infinity).

The non-negativity of $f(t)$ at $t = 0$ results in the first condition and the non-negativity of $f(t)$ at $t \to \infty$ results in the second condition of the theorem.

In the rest we suppose that $a_1 > 0$ and $a_1 + a_2 + a_3 \geq 0$. We investigate the non-negativity of $f(t)$ by constructing $f^*(t) = a_1 e^{-\lambda_1 t} + a_2 e^{-\lambda_2 t} + a_3^* e^{-\lambda_3 t}$ such that a_3^* takes the minimal a_3 value with which $f(t)$ is still non-negative, i.e., we will have $f^*(c) = 0$ for some $c \geq 0$.

We have the following two cases:

a) $f^*(c)$ touches the x-axes at $c > 0$, that is, $f^*(c) = 0$ and $f'^*(c) = 0$,
b) $f^*(0) = 0$ and $f'^*(0) \geq 0$.

In case a) we have

$$f^*(c) = a_1 e^{-\lambda_1 c} + a_2 e^{-\lambda_2 c} + a_3^* e^{-\lambda_3 c} = 0, \tag{7}$$

$$f'^*(c) = -a_1\lambda_1 e^{-\lambda_1 c} - a_2\lambda_2 e^{-\lambda_2 c} - a_3^*\lambda_3 e^{-\lambda_3 c} = 0, \tag{8}$$

from which

$$\frac{a_2}{a_1} = -\frac{\lambda_3 - \lambda_1}{\lambda_3 - \lambda_2}\, e^{(\lambda_2-\lambda_1)c}, \tag{9}$$

$$\frac{a_3^*}{a_1} = \frac{\lambda_2 - \lambda_1}{\lambda_3 - \lambda_2}\, e^{(\lambda_3-\lambda_1)c}. \tag{10}$$

If $a_2 \geq -a_1 \frac{\lambda_3-\lambda_1}{\lambda_3-\lambda_2}$ then there is no $c > 0$ that satisfies (9), since the left hand side of (9) is negative and less than $-\frac{\lambda_3-\lambda_1}{\lambda_3-\lambda_2}$. Consequently, case a) is not possible when $a_2 \geq -a_1 \frac{\lambda_3-\lambda_1}{\lambda_3-\lambda_2}$.

If $a_2 < -a_1 \frac{\lambda_3-\lambda_1}{\lambda_3-\lambda_2}$ then c is obtained from (9) as

$$c = \frac{\log\left(-\frac{a_2}{a_1} \frac{\lambda_3-\lambda_2}{\lambda_3-\lambda_1}\right)}{\lambda_2 - \lambda_1},$$

and substituting it to (10) gives

$$a_3^* = a_1 \frac{\lambda_2 - \lambda_1}{\lambda_3 - \lambda_2} \left(-\frac{a_2}{a_1} \frac{\lambda_3 - \lambda_2}{\lambda_3 - \lambda_1} \right)^{\frac{\lambda_3 - \lambda_1}{\lambda_2 - \lambda_1}}.$$

In case b) we have

$$f^*(0) = a_1 + a_2 + a_3^* = 0, \tag{11}$$

$$f'^*(0) = -a_1\lambda_1 - a_2\lambda_2 - a_3^*\lambda_3 \geq 0. \tag{12}$$

Substituting $a_3^* = -a_1 - a_2$ from (11) into (12) we have that (12) holds when $a_2 \geq -a_1 \frac{\lambda_3 - \lambda_1}{\lambda_3 - \lambda_2}$. □

3.2 Case B: 2 Different Real Eigenvalues

In this case we have two options.

- The multiplicity of the dominant eigenvalue, λ_1, $(\lambda_1 < \lambda_2)$ is one and hence the general form of the density function is

$$f_1(t) = a_1 \; e^{-\lambda_1 t} + (a_2 + a_{21} t) \; e^{-\lambda_2 t}. \tag{13}$$

- The multiplicity of the dominant eigenvalue , λ_1, $(\lambda_1 < \lambda_2)$ is two and hence the general form of the density function is

$$f_2(t) = (a_1 + a_{11} t) \; e^{-\lambda_1 t} + a_2 \; e^{-\lambda_2 t}. \tag{14}$$

Theorem 2. *$f_1(t)$ is non-negative for $t \geq 0$ iff*

$$a_1 + a_2 > 0 \quad \textit{and} \quad a_1 \geq 0 \quad \textit{and} \quad a_{21} \geq a_{21}^*$$

where a_{21}^* is that solution of

$$a_{21} \; e^{a_2\lambda_2/a_{21}} + a_1(\lambda_2 - \lambda_1) \; e^{1 + a_2\lambda_1/a_{21}} = 0 \tag{15}$$

which satisfies $a_{21} < a_2(\lambda_2 - \lambda_1)$.

Proof. $f_1(t)$ is a monotone increasing function of a_1, a_2 and a_{21} for $t \geq 0$ and both $f_1(t)$ and $f_1'(t)$ can have at most 2 roots in $(0, \infty)$ (excluding 0 and infinity).

The non-negativity of $f_1(t)$ at $t = 0$ results in the first condition and the non-negativity of $f_1(t)$ at $t \to \infty$ results in the second condition. The minimal a_{21} value for which $f_1(t)$ is non-negative is obtained assuming that $f_1(t)$ touches the x axes at $t = c > 0$, i.e., $f_1(c) = 0$ and $f_1'(c) = 0$. Solving this set of equations for a_{21} and c, we have

$$c = \frac{a_{21} - a_2(\lambda_2 - \lambda_1)}{a_{21}(\lambda_2 - \lambda_1)}, \tag{16}$$

and (15). If $a_{21} > a_2(\lambda_2 - \lambda_1)$ then (16) does not have solution for positive c, i.e., $f_1(t)$ is nonnegative. If $a_{21} < a_2(\lambda_2 - \lambda_1)$ then (16) has a positive solution and a_{21} has to be not less than the associated a_{21}^*. □

Theorem 3. *$f_2(t)$ is non-negative for $t \geq 0$ iff*

$$a_1 + a_2 > 0 \quad \textit{and} \quad a_{11} \geq 0 \quad \textit{and} \quad a_{11} \geq a_{11}^*$$

where a_{11}^* is that solution of

$$a_2 \; e^{\lambda_2 \left(\frac{a_1}{a_{11}} - \frac{1}{\lambda_2 - \lambda_1} \right)} - a_{11}(\lambda_2 - \lambda_1) \; e^{\lambda_1 \left(\frac{a_1}{a_{11}} - \frac{1}{\lambda_2 - \lambda_1} \right)} = 0 \tag{17}$$

which satisfies $a_{11} < -a_1(\lambda_2 - \lambda_1)$.

Proof. The proof follows the same pattern as the one for $f_1(t)$. □

It has to be noted that the third condition of Theorem 2 and 3 are transcendent, and consequently, numerical methods are required to compute them.

3.3 Case C: 1 Real Eigenvalue

In this case the general form of the density function is

$$f(t) = (a_0 + a_1 t + a_2 t^2) \; e^{-\lambda t}. \tag{18}$$

Theorem 4. *$f(t)$ is non-negative for $t \geq 0$ iff*

$$a_0 > 0 \quad \textit{and} \quad a_2 > 0 \quad \textit{and} \quad a_1 \geq -2\sqrt{a_0 a_2}.$$

Proof. $f(t)$ is a monotone increasing function of a_0, a_1 and a_2 for $t \geq 0$ and both $f(t)$ and $f'(t)$ can have at most 2 roots in $(0, \infty)$ (excluding 0 and infinity). The non-negativity of $f(t)$ at $t = 0$ results in the first condition and the non-negativity of $f(t)$ at $t \to \infty$ results in the second condition.

Supposing that $a_0 > 0$ and $a_2 > 0$ we have the following two cases:

- if $a_1 \geq 0$ then $a_0 + a_1 t + a_2 t^2$ is monotone increasing on $(0, \infty)$,
- if $a_1 < 0$ then $a_0 + a_1 t + a_2 t^2$ has a minimum at $t = -\frac{a_1}{2a_2}$ which is $a_0 - \frac{a_1^2}{4a_2}$.

From which the third condition comes. □

3.4 Case D: One Real and a Pair of Complex Eigenvalues

In this case the general form of the density function is

$$f(t) = a_1 \; e^{-\lambda_1 t} + a_2 \cos(\omega t + \phi) \; e^{-\lambda_c t} \tag{19}$$

where for uniqueness, a_2 and ϕ are defined such that $a_2 > 0$ and $-\pi < \phi \leq \pi$.

Theorem 5. *$f(t)$ is non-negative for $t \geq 0$ iff*

- $a_1 + a_2 \cos(\phi) > 0$ *and*
- $a_1 > 0$ *and*
- $\lambda_1 \leq \lambda_c$ *and*

- $a_2 < a_1 e^{(\lambda_c - \lambda_1)\frac{2\pi}{\omega}}$ *and*
- *if* $a_1 < a_2$ $(< a_1 e^{(\lambda_c - \lambda_1)\frac{2\pi}{\omega}})$ *then* $f(\check{t}) \geq 0$ *and* $f(\hat{t}) \geq 0$ *and*
- *if* $a_1 < a_2$ $(< a_1 e^{(\lambda_c - \lambda_1)\frac{2\pi}{\omega}})$ *and* $f'(t)$ *has roots in* $[\check{t}, \hat{t}]$ *then* $f(t) \geq 0$ *at those roots*

where $\check{t} = \max\left(0, \frac{\pi - 2\phi}{2\omega}\right)$ *and* $\hat{t} = \min\left(\frac{1}{\lambda_c - \lambda_1} \log\left(\frac{a_2}{a_1}\right), \frac{\pi - \phi}{\omega}\right)$.

Proof. The non-negativity of $f(t)$ at $t = 0$ results in the first condition and the non-negativity of $f(t)$ at $t \to \infty$ results in the second and the third conditions. The non-negativity of $f(t)$ for $0 < t < \infty$ is determined by two main factors:

- the relation of the two exponential functions $a_1\ e^{-\lambda_1 t}$ and $a_2\ e^{-\lambda_c t}$,
- the value of the periodic term $\cos(\omega t + \phi)$.

From now on we assume that the first 3 conditions hold. The function $f(t)$ has then the following properties:

- $f(t)$ is a monotone increasing function of a_1 for $t \geq 0$.
- Both $f(t)$ and $f'(t)$ might have infinitely many roots in $(0, \infty)$ (excluding 0 and infinity).
- If $a_1 > a_2$ then $a_1\ e^{-\lambda_1 t} > a_2\ e^{-\lambda_c t}$ for $\forall t > 0$, and consequently $f(t) > 0$ for $\forall t > 0$.
- If $a_1 < a_2$ and $\lambda_1 < \lambda_c$ then $a_1\ e^{-\lambda_1 t} > a_2\ e^{-\lambda_c t}$ for $\forall t > t_r = \frac{1}{\lambda_c - \lambda_1} \log(\frac{a_2}{a_1})$, and consequently $f(t) > 0$ for $\forall t > t_r$. This is because for t_r we have $a_1\ e^{-\lambda_1 t_r} = a_2\ e^{-\lambda_c t_r}$ and $\lambda_1 \leq \lambda_c$.
- If at the end of the first period of $\cos(\omega t + \phi)$, i.e., at $t_p = \frac{2\pi}{\omega}$, we have $a_1\ e^{-\lambda_1 t_p} < a_2\ e^{-\lambda_c t_p}$, then for $t = \frac{\pi - \phi}{\omega} < t_p$ we have

$$\begin{aligned} f(\tfrac{\pi-\phi}{\omega}) &= a_1\ e^{-\lambda_1 \frac{\pi-\phi}{\omega}} + a_2 \cos(\omega \tfrac{\pi-\phi}{\omega} + \phi)\ e^{-\lambda_c \frac{\pi-\phi}{\omega}} \\ &= a_1\ e^{-\lambda_1 \frac{\pi-\phi}{\omega}} - a_2\ e^{-\lambda_c \frac{\pi-\phi}{\omega}} < a_1\ e^{-\lambda_1 t_p} - a_2\ e^{-\lambda_c t_p} < 0\ , \end{aligned}$$

  i.e., $f(t)$ is negative for a positive t.
- If $f(t) > 0$ for $\forall t > 0$ for a given a_2, then for $\forall \tilde{a}_2 \in [0, a_2]$ the function $\tilde{f}(t) = a_1\ e^{-\lambda_1 t} + \tilde{a}_2 \cos(\omega t + \phi)\ e^{-\lambda_c t} > 0$ for $\forall t > 0$.
- If $f(t) \geq 0$ for $\forall t \in [0, t_p]$ then $f(t) \geq 0$ for $\forall t > 0$. This is because the non-negativity of $f(t)$ for $[t_p, \infty)$ is equivalent to the non-negativity of $a_1\ e^{-\lambda_1 t} + \tilde{a}_2\ \cos(\omega t + \phi)\ e^{-\lambda_c t}$ where $\tilde{a}_2 = a_2\ e^{-(\lambda_c - \lambda_1) t_p} \leq a_2$.

Based on the above properties, if $\lambda_1 < \lambda_c$,

- $a_2 \leq a_1$ implies that $f(t)$ is non-negative.
- $a_2 > a_1 e^{(\lambda_c - \lambda_1)\frac{2\pi}{\omega}}$ or equivalently $t_r > t_p$ implies that $f(t)$ is negative for positive values of t,
- if $a_1 < a_2 \leq a_1 e^{(\lambda_c - \lambda_1)\frac{2\pi}{\omega}}$ then $f(t)$ can become negative depending on the initial phase of the cosine term. $f(t)$ can become negative only when the cosine term is negative but due to the faster decay of the $e^{-\lambda_c t}$ term it is enough to study the first interval where the cosine term takes negative values,

i.e., $(\frac{\pi-2\phi}{2\omega}, \frac{\pi-\phi}{\omega})$. Depending on the initial phase, ϕ, $\frac{\pi-2\phi}{2\omega}$ can be less than 0 and $\frac{\pi-\phi}{\omega}$ can be greater than t_r. Considering these additional constraints, $\check{t}$ and $\hat{t}$ define the borders of the decisive interval. If $f(t)$ is non-negative on $[\check{t}, \hat{t}]$, it is non-negative for $\forall t > 0$.

We have $f'(\check{t}) < 0$ because both $e^{-\lambda_1 t}$ and $\cos(\omega t + \phi)\ e^{-\lambda_c t}$ decay at $t = \check{t}$. If $f'(\hat{t})$ is non-positive, $f'(t)$ has 0, 1, or 2 roots in $[\check{t}, \hat{t}]$, and the sign of $f(t)$ at these roots decides the non-negativity of $f(t)$. If $f'(\hat{t})$ is positive, $f(t)$ has a single minimum in $[\check{t}, \hat{t}]$, and the sign of this minimum decides the non-negativity of $f(t)$. □

4 Moments Bounds of the ME(3) Class

The previous section provides results to check the ME(3) membership of order 3 matrix exponential functions. We implemented those checks in a Mathematica function. Using this implementation, in this section, we numerically investigate the flexibility of the ME(3) class compared to the limits of the PH(3) class, for which similar results are provided in [6] to check PH(3) membership.

A continuous ME(3) or PH(3) distribution is uniquely characterized by its first 5 moments. For a given set of $\{\mu_1, \ldots, \mu_5\}$ moments we check the ME(3) and PH(3) membership with a two step procedure.

- The first step is to compute a vector and matrix pair of order 3, $(\underline{v}, \boldsymbol{H})$, for which $i!\underline{v}(-\boldsymbol{H})^{-i}\mathbb{1} = \mu_i, i = 1, \ldots, 5$. The procedure of Appie van de Liefvoort in [12] provides such $(\underline{v}, \boldsymbol{H})$ pair with a proper transformation of the closing vector.[1]
- Starting from $(\underline{v}, \boldsymbol{H})$, if the PH(3) transformation procedure in [6] generates a valid canonical representation then $\{\mu_1, \ldots \mu_5\}$ represents a member of the PH(3) set. Similarly, if the matrix exponential function, $\underline{v}e^{\boldsymbol{H}t}(-\boldsymbol{H})\mathbb{1}$, is non-negative according to the checks of the previous section then $\{\mu_1, \ldots \mu_5\}$ represents a member of the ME(3) set.

As in previous works, to reduce the number of parameters we introduce the normalized moments, $n_i = \frac{\mu_i}{\mu_{i-1}\mu_1}$, which eliminate a scaling factor and represent the shape of ME(3) and PH(3) distributions with 4 parameters, $\{n_2, n_3, n_4, n_5\}$.

The subsequent numerical results are divided into investigations of the n_2, n_3 domain with arbitrary n_4, n_5 and investigations of the n_4, n_5 domain with given n_2, n_3.

4.1 The Second and Third Normalized Moments

The n_2, n_3 normalized moment bounds of the PH(3) class are not completely known yet. There is a proved result for the valid range of the APH(3) class [3],

[1] In [12] the initial and the closing vectors are $\{1, 0, 0, \ldots, 0\}$. In our case the closing vector is $\{1, 1, \ldots, 1\}$, hence a similarity transformation is required as described in [11].

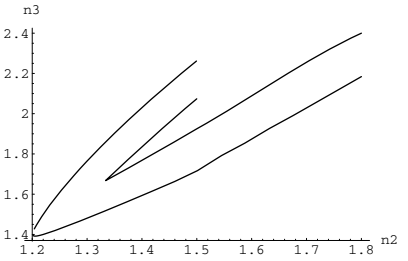


Fig. 1. The range of second and third normalized moments of the PH(3) and ME(3) classes

and there is a numerically checked conjecture that the related borders of the PH(3) class coincide with the ones of the APH(3) class [6]. Here we compare the borders of the ME(3) class with these borders of the PH(3) class.

To check if an n_2, n_3 pair is inside the range of the ME(3) class is rather difficult. We have tools to check if $\{n_2, n_3, n_4, n_5\}$ defines an ME(3) distribution. Based on this tool, for a given n_2, n_3 pair a natural procedure would be to check the ME(3) membership of $\{n_2, n_3, x, y\}$, where x and y run through the positive quarter plain. Unfortunately, this procedure is infeasible, because it is practically impossible to find valid n_4, n_5 pairs with exhaustive search.

To get around this problem we applied special ME(3) subclasses whose structure is defined by 2 shape parameters and a scaling factor. Having these subclasses we set the 2 shape parameters to match n_2, n_3 and checked if we obtained a valid distribution.

The Exp-Erlang and the Erlang-Exp distributions in [3] form such subsets, which we used for n_2, n_3 pairs inside the range of the PH(3) class.

For n_2, n_3 pairs outside the range of the PH(3) class we used the following function with complex roots ($a_1 = a_2 = a$, $\lambda_1 = \lambda_2 = \lambda$ in (19))

$$f(t) = a\ e^{-\lambda t}(1 + \cos(\omega t + \phi)) \tag{20}$$

where a is a normalizing constant ($\int_t e^{-\lambda t}(1 + \cos(\omega t + \phi))dt = 1/a$), λ is the scaling factor and ω and ϕ are the two shape parameters. When $\lambda = 1$

$$n_2 = \frac{2\left(\sqrt{1+\omega^2} + \cos(\phi + \arctan(\omega))\right)\left(\left(1+\omega^2\right)^{\frac{3}{2}} + \cos(\phi + 3\arctan(\omega))\right)}{\left(1 + \omega^2 + \cos(\phi + 2\arctan(\omega))\right)^2},$$

$$n_3 = \frac{3\left(\sqrt{1+\omega^2} + \cos(\phi + \arctan(\omega))\right)\left(\left(1+\omega^2\right)^2 + \cos(\phi + 4\arctan(\omega))\right)}{\left(1 + \omega^2 + \cos(\phi + 2\arctan(\omega))\right)\left(\left(1+\omega^2\right)^{\frac{3}{2}} + \cos(\phi + 3\arctan(\omega))\right)}.$$

For a given n_2, n_3 pair solving this equation for the ϕ, ω pair gives a matrix exponential function whose second and third normalized moments are n_2 and n_3. The non-negativity of this function can be checked by Theorem 5.

Figure 1 depicts the borders of the ME(3) class (obtained for subclass (20)) and the borders of the PH(3) class (inner borders of the figure) on the n_2, n_3

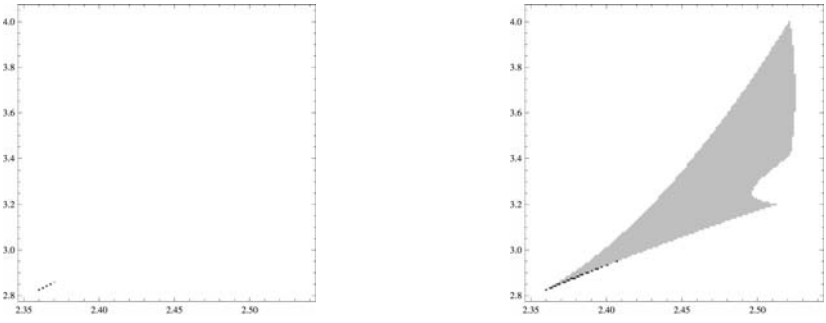


Fig. 2. Realizable n_4, n_5 normalized moments with PH(3) (on the left) and ME(3) (on the right) in case of $n_2 = 1.45$ and $n_3 = 1.9015$

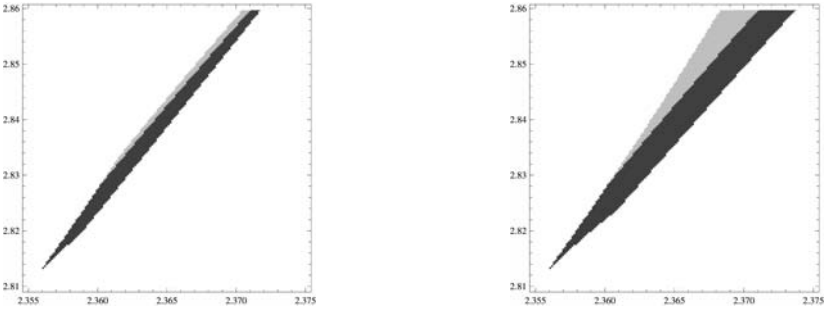


Fig. 3. Lower peak of the realizable n_4, n_5 region with PH(3) (on the left) and ME(3) (on the right) in case of $n_2 = 1.45$ and $n_3 = 1.9015$

plain. Our numerical investigations suggest that the outer borders in Figure 1 are the borders of the whole ME(3) class, but we cannot prove it. The left most point of these borders, $n_2 = 1.200902$ gives the ME(3) distribution with minimal n_2 or, equivalently, with minimal coefficient of variation, and this point corresponds to the minimal coefficient of variation of the ME(3) class reported in [4]. The PH(3) class, and consequently the ME(3) class, are known to be only lower bounded when $n_2 > 1.5$. That is why the upper bound curves end at $n_2 = 1.5$.

The results of Section 3 indicate already that the borders of the ME(3) class do not exhibit nice closed form expressions, but numerical methods are required for their evaluation. We used the standard floating point precision of Mathematica to compute the presented results, but these computations are numerically sensitive.

4.2 The Fourth and Fifth Normalized Moments

In this section we study the region of realizable fourth and fifth normalized moments (n_4 and n_5) for a given pair of second and third normalized moments (n_2 and n_3). In order to find this region we make use of the subclasses presented

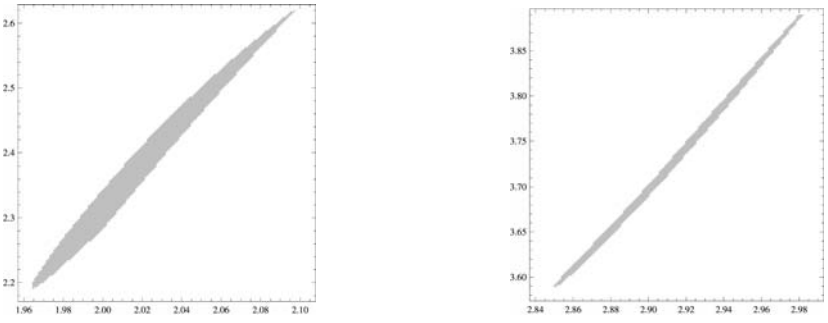


Fig. 4. Realizable n_4, n_5 normalized moments with ME(3) for $n_2 = 1.45$ and $n_3 = 1.725$ (on the left) and $n_2 = 1.45$ and $n_3 = 2.1$ (on the right)

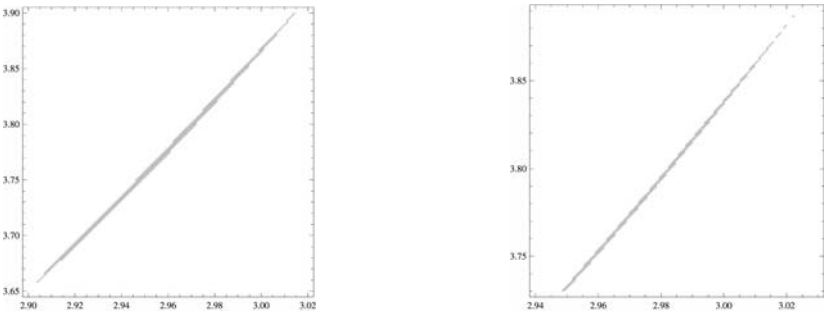


Fig. 5. Realizable n_4, n_5 normalized moments with ME(3) for $n_2 = 1.45$ and $n_3 = 2.1249$ (on the left) and $n_2 = 1.45$ and $n_3 = 2.1373$ (on the right)

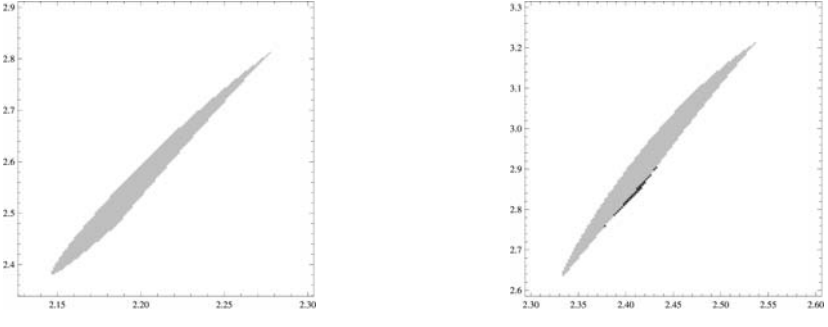


Fig. 6. Realizable n_4, n_5 normalized moments with ME(3) for $n_2 = 1.6$ and $n_3 = 1.9$ (on the left) and $n_2 = 1.6$ and $n_3 = 2.0$ (on the right)

in Section 4.1. We use Erlang-Exp distributions [3] inside the PH(3) borders of Figure 1 and the subclass defined by (20) between the PH(3) and the ME(3) borders. First we generate a matrix exponential function from the given ME(3) subclass that realizes the pair (n_2,n_3). Then we calculate n_4 and n_5 for this

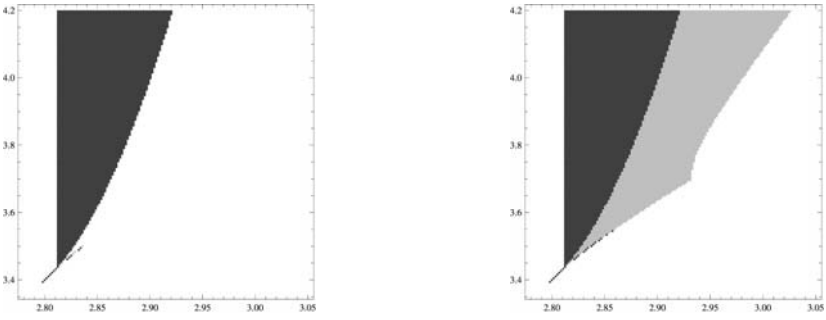


Fig. 7. Lower peak of the realizable n_4, n_5 region with PH(3) (on the left) and ME(3) (on the right) in case of $n_2 = 1.6$ and $n_3 = 2.2$

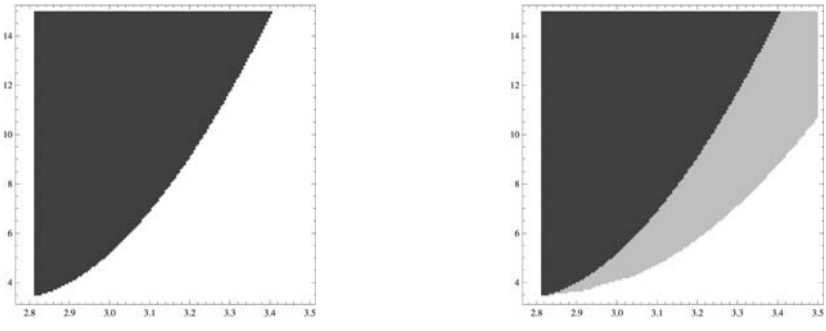


Fig. 8. Realizable n_4, n_5 region with PH(3) (on the left) and ME(3) (on the right) in case of $n_2 = 1.6$ and $n_3 = 2.2$

matrix exponential function and use them as starting point in exploring the realizable region of n_4 and n_5. Since the realizable region of the PH(3) class is a subregion of the realizable region of the ME(3) class, it is easier to start from a PH(3) point if possible.

We start by considering cases for which $n_2 = 1.45$. Based on the results presented in Section 4.1, with this value of n_2 the interval of realizable third normalized moments is $(1.6517, 2.1498)$ with ME(3) while it is $(1.8457, 1.9573)$ with PH(3). First we look at the middle point of the n_3 interval that can be realized with a PH(3), i.e., $n_3 = 1.9015$. Figure 2 depicts the realizable region of n_4 and n_5 for both PH(3) and ME(3). In all the figures we have n_4 on the x-axes and n_5 on the y-axes. Further, the lighter gray region contains the points that are realized with a ME(3) or PH(3) with one real and a pair of complex eigenvalues (class D) while the darker gray area contains points where the ME(3) or PH(3) is realized with three real eigenvalues. It is clear from Figure 2 that the ME(3) gives much higher flexibility than the PH(3) does. In Figure 3 we concentrate on the lower peak of the regions depicted in Figure 2. ME(3) is somewhat more flexible in this subregion as well and one can observe that the

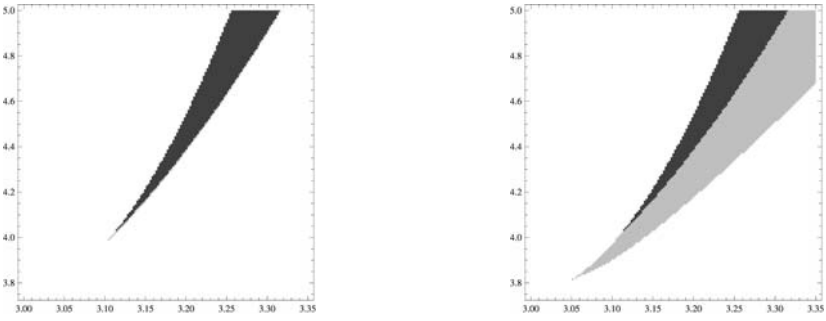


Fig. 9. Realizable n_4, n_5 region with PH(3) (on the left) and ME(3) (on the right) in case of $n_2 = 1.6$ and $n_3 = 2.3$

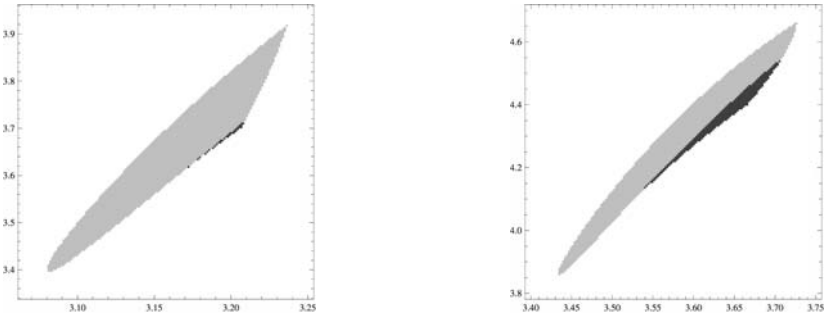


Fig. 10. Realizable n_4, n_5 region with ME(3) for $n_3 = 2.7333$ (on the left) and $n_3 = 2.9333$ (on the right) in case of $n_2 = 2.6$

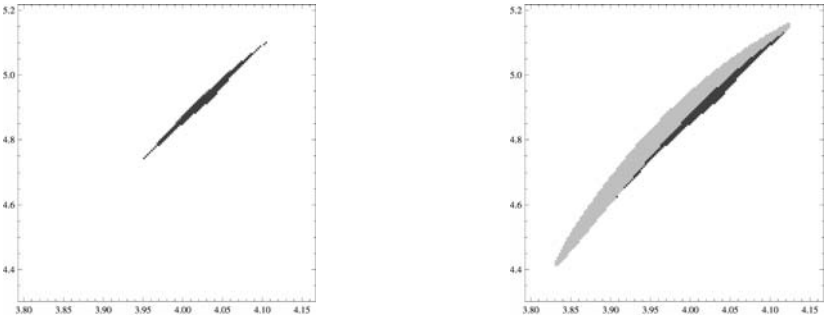


Fig. 11. Realizable n_4, n_5 region with PH(3) (on the left) and ME(3) (on the right) in case of $n_2 = 2.2$ and $n_3 = 3.1333$

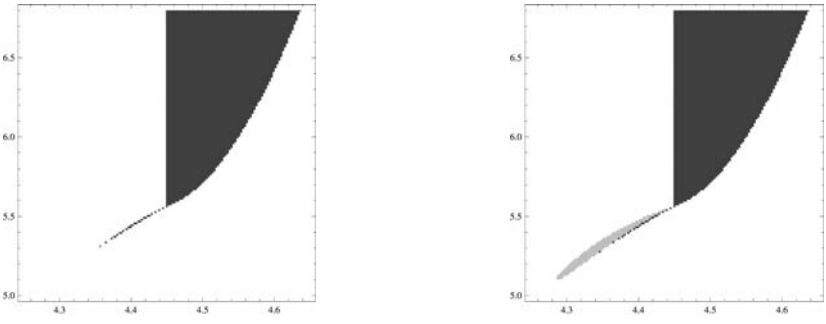


Fig. 12. Realizable n_4, n_5 region with PH(3) (on the left) and ME(3) (on the right) in case of $n_2 = 2.2$ and $n_3 = 3.3333$

flexibility is increased both for what concerns the distribution with one real and two complex eigenvalues and for what concerns the distributions with three real eigenvalues.

Now we turn our attention to such n_3 values that cannot be realized by a PH(3) with $n_2 = 1.45$. In particular, Figure 4 depicts the realizable n_4,n_5 regions for $n_2 = 1.45$,$n_3 = 1.725$ and $n_2 = 1.45$,$n_3 = 2.1$ which lie respectively beneath and above the n_3 interval that can be realized with PH(3). By comparison with Figure 2 it is clear that approaching the possible minimum and maximum values of n_3 the realizable n_4,n_5 region not only changes its shape but it is shrinking as well. To illustrate further this shrinking, Figure 5 depicts the realizable n_4,n_5 region for $n_2 = 1.45$,$n_3 = 2.1249$ and $n_2 = 1.45$,$n_3 = 2.1373$ where the realizable region gets narrower and shorter.

Next we investigate a few cases with $n_2 = 1.6$. We start with two such values of n_3, namely 1.9 and 2.0, that cannot be realized with a PH(3). The realizable n_4, n_5 pairs are depicted in Figure 6. Diverging from the minimal n_3 value, i.e. by increasing the actual value of n_3 the realizable region becomes larger. Diverging further from the minimal n_3 value, we choose $n_3 = 2.2$ which can be realized by PH(3). Figure 7 depicts the lower peak of the realizable n_4, n_5 region for PH(3) and ME(3). This figure reports new qualitative properties. It indicates that the realizable n_4, n_5 region can be composed by more than two areas and the areas are not concave. The $n_2 = 1.6$, $n_3 = 2.2$ case is further illustrated by Figure 8, there is no upper bound for n_4 and n_5. Figure 9 illustrates instead how the realizable n_4,n_5 is changed and moved by increasing n_3 to 2.3.

In the following we investigate cases with $n_2 = 2.2$. Figure 10 depicts the realizable region for $n_3 = 2.7333$ which cannot be realized by PH(3) and $n_3 = 2.9333$ which is the lower limit for PH(3), i.e., in this point a single (n_4, n_5) point can be realized with PH(3). For $n_3 = 3.1333$ the regions are shown in Figure 11 and for $n_3 = 3.3333$ in Figure 12. With $n_3 = 3.1333$ there are upper bounds for n_4 and n_5 which are not present with $n_3 = 3.3333$.

5 Conclusions

This paper is devoted to the investigation of the border of ME(3) distributions. To this end we collected necessary and sufficient conditions for different kinds of order 3 matrix exponential functions to be non-negative. It turned out that these conditions are explicit in some cases, but they require the solution of a transcendent equation in other cases. Due to this fact, only numerical methods are available for the investigation of ME(3) borders.

Using those necessary and sufficient conditions we completed a set of numerical evaluations. The results show, in accordance with the common expectations, that the ME(3) set has very complex moments borders and it is significantly larger than the PH(3) set.

References

1. Asmussen, S., O'Cinneide, C.A.: Matrix-exponential distributions – distributions with a rational Laplace transform. In: Kotz, S., Read, C. (eds.) Encyclopedia of Statistical Sciences, pp. 435–440. John Wiley & Sons, New York (1997)
2. Bean, N.G., Nielsen, B.F.: Quasi-birth-and-death processes with rational arrival process components. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, IMM-Technical Report-2007-20 (2007)
3. Bobbio, A., Horváth, A., Telek, M.: Matching three moments with minimal acyclic phase type distributions. Stochastic models, 303–326 (2005)
4. Éltető, T., Rácz, S., Telek, M.: Minimal coefficient of variation of matrix exponential distributions. In: 2nd Madrid Conference on Queueing Theory, Madrid, Spain (July 2006) (abstract)
5. Fackrell, M.W.: Characterization of matrix-exponential distributions. Technical report, School of Applied Mathematics, The University of Adelaide, Ph. D. thesis (2003)
6. Horváth, G., Telek, M.: On the canonical representation of phase type distributions. Performance Evaluation (2008), doi:10.1016/j.peva.2008.11.002
7. Latouche, G., Ramaswami, V.: Introduction to matrix analytic methods in stochastic modeling. SIAM, Philadelphia (1999)
8. Lipsky, L.: Queueing Theory: A linear algebraic approach. MacMillan, New York (1992)
9. Neuts, M.: Matrix-Geometric Solutions in Stochastic Models. John Hopkins University Press, Baltimore (1981)
10. Pérez, J.F., Van Velthoven, J., Van Houdt, B.: Q-mam: A tool for solving infinite queues using matrix-analytic methods. In: Proceedings of SMCtools 2008, Athens, Greece. ACM Press, New York (2008)
11. Telek, M., Horváth, G.: A minimal representation of Markov arrival processes and a moments matching method. Performance Evaluation 64(9-12), 1153–1168 (2007)
12. van de Liefvoort, A.: The moment problem for continuous distributions. Technical report, University of Missouri, WP-CM-1990-02, Kansas City (1990)

Analysis of Discrete-Time Buffers with General Session-Based Arrivals

Sabine Wittevrongel, Stijn De Vuyst, and Herwig Bruneel

SMACS Research Group*, Department of Telecommunications and Information Processing (TELIN), Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
Tel.: +32-9-264 89 01, Fax: +32-9-264 42 95
{sw,sdv,hb}@telin.ugent.be

Abstract. Session-based arrival streams are a new approach for modelling the traffic generated by users in a telecommunication network. In this paper, we analyze the behavior of a discrete-time buffer with one output line, an infinite storage capacity and session-based arrivals. Users from an infinite user population can start and end sessions during which they are active and send packets to the buffer. Each active user generates a random but strictly positive number of packets per time slot. Unlike in previous work, there are T different session types and for each type, the session-length distribution is general. The resulting discrete-time queueing model is analyzed by means of an analytical technique, which is basically a generating-functions approach that uses an infinite-dimensional state description. Expressions are obtained for the steady-state probability generating functions of both the buffer content and the packet delay. From these, the mean values and the tail distributions of the buffer content and the packet delay are derived as well. Some numerical examples are shown to illustrate the influence of the session-based packet arrival process on the buffer behavior.

Keywords: Discrete-time queueing model, Session-based arrivals, General session lengths, Analytic study, Buffer content and delay.

1 Introduction

In many subsystems of packet-based telecommunication networks, buffers are used for the temporary storage of information packets. In order to assess the behavior of these buffers, appropriate traffic models need to be considered. In particular, there is a continuing need for models that can accurately capture the correlated nature of the traffic streams in modern telecommunication networks. Session-based arrival streams are a new traffic modelling approach. We consider an infinite user population where each user can start and end sessions. During a session a user is active and sends information packets through the communication system. Since we focus on discrete-time models, we assume time is divided

* SMACS: Stochastic Modeling and Analysis of Communication Systems.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 189–203, 2009.

into fixed-length slots. Each active user generates a random but strictly positive number of packets per slot. Note that such session-based packet generation introduces time correlation in the packet arrival process. Session-based arrivals are illustrated in Fig. 1. Here the term session length denotes the number of consecutive slots during which a user remains active, whereas the number of packets generated per slot during a session is referred to as the session bandwidth.

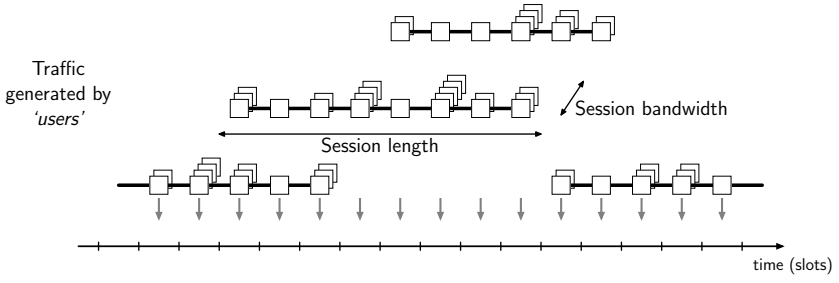


Fig. 1. Session-based packet arrivals: session length and bandwidth

A possible application of session-based arrival processes is depicted in Fig. 2. A web server accepts requests from users for a certain web page or file and responds by sending the requested file to the user. The web server is connected to the internet through a gateway and this gateway contains a buffer for outgoing data from the server to the internet. If we define the download of a file by a user as one session, the traffic towards the output buffer of the web server can be adequately described by a session-based arrival process.

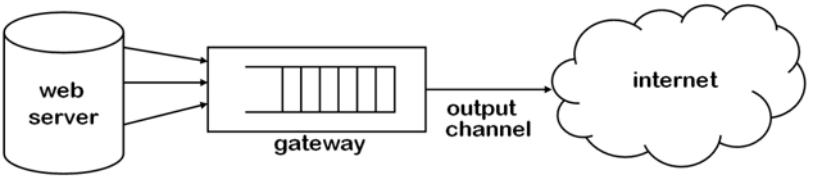


Fig. 2. A web server connected to the internet through a gateway

In previous work [10,11], we have analyzed a discrete-time queue with session-based arrivals and geometrically distributed session lengths. The train arrival process, where messages (the equivalent of what we consider sessions) arrive to the queue at the rate of exactly one packet per slot, is considered in [3,5,7,14,15,16]. Also somewhat related are the on/off-type arrival models studied in [8,12,18], where a finite number of users generate one packet per slot during on-periods and no packets during off-periods. In [6], messages consisting of a fixed number of packets are considered in case of an uncorrelated packet arrival process. A related continuous-time model is analyzed in [1].

The aim of the present paper is to further extend the previous analyses to a discrete-time queueing system with general session-based arrivals. Specifically, unlike in previous work, we consider *heterogeneous* sessions of T different types with *general* type-dependent session-length distributions. This extension allows e.g. to take into account the fact that files on a web server are typically either small or very large [2]. A model with generally distributed session lengths moreover makes it possible to investigate the impact of the nature of the session-length distributions on the buffer behavior. We develop a mathematical analysis technique, that makes extensive use of probability generating functions (PGFs). As opposed to our previous work for geometric session lengths (see [10,11]), an infinite-dimensional state description is required in case of generally distributed session lengths, which seriously complicates the analysis.

The outline of the paper is as follows. In Sect. 2, we describe the queueing model under study. In Sect. 3, a set of state variables is defined and the system equations are established. A functional equation for the joint PGF of the system state vector is obtained in Sect. 4. Some further characteristics of the session-based packet arrival process are studied in Sect. 5. Section 6 concentrates on the derivation of the mean value, the PGF and the tail distribution of the buffer content from the functional equation. The packet delay is analyzed in Sect. 7, for a first-come-first-served (FCFS) queueing rule for packets. Some numerical examples are discussed in Sect. 8. Finally, the paper is concluded in Sect. 9.

2 Queueing Model Description

We study a discrete-time queueing system with one single output line and an infinite storage capacity for packets. As usual for discrete-time models (see e.g. [4,13]), the time axis is divided into fixed-length slots and transmissions from the buffer can only start at slot boundaries. Therefore, when the queueing system is empty at the beginning of a slot, no packet can leave the buffer at the end of that slot, even if some packets have arrived to the buffer during the slot.

A session-based arrival process is considered. Users from an infinite user population can start and end sessions during which they are active and send packets to the queueing system. When a user starts a session, he generates a random but strictly positive number of packets per slot. The session ends when the user has no more data left to send.

There are T different session types. For sessions of type t $(1 \leq t \leq T)$, the session lengths (expressed in slots) are assumed to be independent and identically distributed (i.i.d.) random variables with the following probability mass function (PMF) and PGF:

$$\ell_t(i) = \text{Prob}[\text{session length of type } t \text{ is } i \text{ slots}] \ , \quad i \geq 1 \ ; \tag{1}$$

$$L_t(z) = \sum_{i=1}^{\infty} \ell_t(i)\, z^i \ . \tag{2}$$

The numbers of new sessions of type t started by the user population during the consecutive slots are assumed to be i.i.d. random variables with common PGF $S_t(z)$. Since in normal conditions, internet users act independently from each other, this seems like a realistic assumption. The numbers of packets generated per slot during a session of type t are assumed to be i.i.d. with PGF $P_t(z)$, where $P_t(0)$ equals zero, since at least one packet is generated per slot per session. Sessions of different types are assumed to be independent.

The queueing system has an unreliable output line subject to random failures that are assumed to occur independently from slot to slot. The output line availability is modelled by a parameter σ. Specifically, σ is the probability that the output line is available during a slot. The transmission times of the packets from the buffer are assumed to be constant, equal to one slot per packet. So, whenever the queueing system is nonempty at the beginning of a slot, a packet will leave the buffer at the end of this slot with probability σ and no packet will leave with probability $1-\sigma$, independently from slot to slot. Note that these assumptions result in a geometric distribution (with parameter $1-\sigma$) for the effective transmission times required for the successful transmission of a packet from the queueing system and the mean effective transmission time of a packet equals $1/\sigma$.

3 System Equations

The goal of this section is to introduce a Markovian state description for the queueing system described above. In order to do so, we first take a closer look at the packet arrival process. Let us define $s_k(t)$ as the number of new sessions of type t started during slot k. In view of the model description of Sect. 2, for a given t, the random variables $s_k(t)$ are i.i.d. with common PGF $S_t(z)$. Let $a_{n,k}(t)$ be the random variable representing the number of active sessions of type t that are already active for exactly n slots during slot k. Then the following relationships hold:

$$a_{1,k}(t) = s_k(t) \ ; \tag{3}$$

$$a_{n,k}(t) = \sum_{i=1}^{a_{n-1,k-1}(t)} c^i_{n-1,k}(t) \,, \quad n > 1 \ . \tag{4}$$

The random variable $c^i_{n-1,k}(t)$ in (4) takes on the values 0 or 1, and equals 1 if and only if the ith active session of type t that was in its $(n-1)$th slot during slot $k-1$, remains active in slot k. We define $\pi_t(n-1)$ as the probability that a session of type t that was $n-1$ slots long continues by at least one more slot, i.e.,

$$\pi_t(n-1) \triangleq \frac{1 - \sum_{i=1}^{n-1} \ell_t(i)}{1 - \sum_{i=1}^{n-2} \ell_t(i)} \ . \tag{5}$$

Hence, we have that for given n and t, the $c^i_{n-1,k}(t)$'s are i.i.d. random variables with common PGF

$$C_{n-1,t}(z) \triangleq E\left[z^{c^i_{n-1,k}(t)}\right] = 1 - \pi_t(n-1) + \pi_t(n-1)\, z\,, \quad n > 1\ , \tag{6}$$

where $E[.]$ is the expected value of the argument between square brackets.

Next, let m_k denote the total number of packets generated during slot k. Then m_k can be expressed as

$$m_k = \sum_{t=1}^{T} \sum_{n=1}^{\infty} \sum_{i=1}^{a_{n,k}(t)} p^i_{n,k}(t)\ , \tag{7}$$

where $p^i_{n,k}(t)$ represents the number of packets generated during slot k by the ith session of type t that is already active for exactly n slots. From Sect. 2, it follows that for a given t, the random variables $p^i_{n,k}(t)$ are i.i.d. with PGF $P_t(z)$.

Finally, let u_k denote the buffer content (i.e., the total number of packets in the queueing system, including the packet in transmission, if any) after slot k. The evolution of the buffer content is governed by the following system equation:

$$u_k = (u_{k-1} - r_k)^+ + m_k\ , \tag{8}$$

where $(.)^+ = \max(.,0)$ and the r_k's are i.i.d. Bernoulli random variables equal to zero with probability $1-\sigma$ and equal to one with probability σ, in view of the random interruptions of the output line.

From the above system equations (3)–(8) it is easily seen that the set of vectors $\left\{\left(\underline{\mathbf{a}}_{1,k}, \ldots, \underline{\mathbf{a}}_{T,k}, u_k\right)\right\}$, where $\underline{\mathbf{a}}_{t,k} = (a_{1,k}(t), a_{2,k}(t), \ldots)$, constitutes a Markov chain. The state of the queueing system after slot k can hence be fully described by the infinite-dimensional vector $\left(\underline{\mathbf{a}}_{1,k}, \ldots, \underline{\mathbf{a}}_{T,k}, u_k\right)$.

4 Functional Equation

We start the analysis of the buffer behavior by defining the joint PGF of the state vector $\left(\underline{\mathbf{a}}_{1,k}, \ldots, \underline{\mathbf{a}}_{T,k}, u_k\right)$:

$$Q_k(\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_T, z) \triangleq E\left[\left(\prod_{t=1}^{T} \prod_{n=1}^{\infty} {x_{n,t}}^{a_{n,k}(t)}\right) z^{u_k}\right]\ , \tag{9}$$

where $\underline{\mathbf{x}}_t = (x_{1,t}, x_{2,t}, \ldots)$. With this definition, (7) and (8), $Q_k(\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_T, z)$ can then be obtained as

$$Q_k(\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_T, z) = E\left[\left(\prod_{t=1}^{T} \prod_{n=1}^{\infty} \left(x_{n,t}\, P_t(z)\right)^{a_{n,k}(t)}\right) z^{(u_{k-1}-r_k)^+}\right]\ .$$

Next, by using (3) and (4), and by averaging over the distributions of the $c^i_{n-1,k}(t)$'s, defined in (6), we can transform the expression for $Q_k(\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_T, z)$ further into

$$Q_k(\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_T,z) = \left(\prod_{t=1}^{T} S_t(x_{1,t}\,P_t(z))\right) \cdot E\left[\left(\prod_{t=1}^{T}\prod_{n=1}^{\infty} G_{n,t}(\underline{\mathbf{x}}_t,z)^{a_{n,k-1}(t)}\right) z^{(u_{k-1}-r_k)^+}\right] \quad , \tag{10}$$

where

$$G_{n,t}(\underline{\mathbf{x}}_t,z) \triangleq C_{n,t}(x_{n+1,t}\,P_t(z)) \;, \quad n \geq 1 \;, \quad 1 \leq t \leq T \;. \tag{11}$$

In order to remove the operator $(.)^+$, we need to distinguish between the case where $r_k = 0$, the case where $r_k = 1$, $u_{k-1} > 0$ and the case where $r_k = 1$, $u_{k-1} = 0$. Moreover, we note that $u_{k-1} = 0$ implies that no packets have arrived during slot $k-1$, and hence $a_{n,k-1}(t) = 0$ ($n \geq 1$, $1 \leq t \leq T$), owing to the fact that a packet can never leave the buffer at the end of its arrival slot. With this property, the right-hand side of (10) can be further expressed in terms of the Q_{k-1}-function.

We now assume that the equilibrium condition is satisfied so that the queueing system can reach a steady state. In the steady state, $Q_k(\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_T,z)$ becomes independent of k. As a result, we then obtain the following functional equation for the steady-state PGF $Q(\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_T,z)$:

$$z\,Q(\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_T,z) = \left(\prod_{t=1}^{T} S_t(x_{1,t}\,P_t(z))\right) \cdot \{\Phi(z)\,Q(\underline{\mathbf{G}}_1(\underline{\mathbf{x}}_1,z),\ldots,\underline{\mathbf{G}}_T(\underline{\mathbf{x}}_T,z),z) + \sigma\,(z-1)\,p_0\} \quad , \tag{12}$$

where

$$\underline{\mathbf{G}}_t(\underline{\mathbf{x}}_t,z) \triangleq (G_{1,t}(\underline{\mathbf{x}}_t,z), G_{2,t}(\underline{\mathbf{x}}_t,z),\ldots) \;, \quad 1 \leq t \leq T \;; \tag{13}$$

$$\Phi(z) \triangleq \sigma + (1-\sigma)\,z \;, \tag{14}$$

and p_0 is the steady-state probability of an empty buffer. Note that $\underline{\mathbf{G}}_t(\underline{\mathbf{x}}_t,z)$ only depends on $\underline{\mathbf{x}}_t$ and z, which is due to the fact that sessions of different types are assumed to be independent. In principle, (12) fully describes the steady-state buffer behavior. In the next sections, we will use (12) to derive several explicit results.

5 Packet Arrival Process

First, we study some further characteristics of the session-based packet arrival process. These will prove to be useful for the buffer-content analysis further in the paper. Let $a_n(t)$ denote the steady-state version of $a_{n,k}(t)$. The joint PGF $A(\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_T)$ of the $a_n(t)$'s is then given by $Q(\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_T,1)$. Putting $z = 1$ in (12), we obtain

$$A(\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_T) = \left(\prod_{t=1}^{T} S_t(x_{1,t})\right) A(\underline{\mathbf{G}}_1(\underline{\mathbf{x}}_1,1),\ldots,\underline{\mathbf{G}}_T(\underline{\mathbf{x}}_T,1)) \quad . \tag{15}$$

Successive applications of (15) then lead to

$$A(\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_T) = \prod_{t=1}^{T} \prod_{j=0}^{\infty} S_t \left(\sum_{i=1}^{j} \ell_t(i) \, (1 - x_{j+1,t}) + x_{j+1,t} \right) \quad . \tag{16}$$

Here we have used the definitions (11) and (13) and the following relationships:

$$C_{1,t}(C_{2,t}(\ldots C_{j,t}(x_{j+1,t}) \ldots)) = \sum_{i=1}^{j} \ell_t(i) \, (1 - x_{j+1,t}) + x_{j+1,t} \quad ; \tag{17}$$

$$\lim_{j \to \infty} C_{n,t}(C_{n+1,t}(\ldots C_{j,t}(x_{j+1,t}) \ldots)) = 1 \, , \quad n \geq 1 \quad , \tag{18}$$

which can be derived from (5) and (6). The marginal PGF $A_{n,t}(z)$ of $a_n(t)$ can be obtained from (16) as

$$A_{n,t}(z) = S_t \left(\sum_{i=1}^{n-1} \ell_t(i) \, (1 - z) + z \right) \quad . \tag{19}$$

The average number of sessions of type t that are in their nth slot during an arbitrary slot is then given by

$$E[a_n(t)] = A'_{n,t}(1) = S'_t(1) \left(1 - \sum_{i=1}^{n-1} \ell_t(i) \right) \quad , \tag{20}$$

i.e., the mean number of new sessions of type t started during a slot times the probability of having a session length of at least n slots.

Let m denote the total number of packet arrivals during an arbitrary slot in the steady state. Then the PGF $M(z)$ of m can be derived from (16) as

$$M(z) = A(\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_T)|_{x_{n,t} = P_t(z), \, n \geq 1, \, 1 \leq t \leq T} \quad . \tag{21}$$

The mean number of packet arrivals per slot is then obtained as

$$E[m] = M'(1) = \sum_{t=1}^{T} S'_t(1) \, L'_t(1) \, P'_t(1) \quad . \tag{22}$$

The equilibrium condition states that the load ρ of the queueing system has to be strictly smaller than one:

$$\rho = \frac{M'(1)}{\sigma} < 1 \quad . \tag{23}$$

6 Buffer Content

In this section, we focus on the buffer content u after a slot in the steady state. Starting from the functional equation (12), we derive expressions for the mean value, the PGF and the tail distribution of the buffer content.

6.1 Mean Buffer Content

We can find the mean buffer content if we consider those values of $x_{n,t}$ ($n \geq 1$, $1 \leq t \leq T$) and z for which the arguments of the Q-functions on both sides of (12) are equal to each another, i.e., for which

$$x_{n,t} = G_{n,t}(\underline{\mathbf{x}}_t, z) \quad , \tag{24}$$

or more explicitly,

$$x_{n,t} = 1 - \pi_t(n) + \pi_t(n)\, x_{n+1,t}\, P_t(z) \quad . \tag{25}$$

These relationships can be solved for the $x_{n,t}$'s in terms of z. Denoting the solution for $x_{n,t}$ by $X_{n,t}(z)$, we obtain

$$X_{n,t}(z) = \frac{\sum_{j=n}^{\infty} \ell_t(j)\, P_t(z)^{j-n}}{1 - \sum_{j=1}^{n-1} \ell_t(j)} , \quad n \geq 1 , \quad 1 \leq t \leq T \quad . \tag{26}$$

Note in particular that

$$X_{1,t}(z) = \frac{L_t(P_t(z))}{P_t(z)} \tag{27}$$

and

$$X_{n,t}(1) = 1 , \quad n \geq 1 \quad . \tag{28}$$

Choosing $x_{n,t} = X_{n,t}(z)$ ($n \geq 1$, $1 \leq t \leq T$) in (12), we then get a linear equation for the function $Q(\underline{\mathbf{X}}_1(z), \ldots, \underline{\mathbf{X}}_T(z), z)$, which has the following solution:

$$Q(\underline{\mathbf{X}}_1(z), \ldots, \underline{\mathbf{X}}_T(z), z) = \frac{\sigma\,(z-1)\, p_0\, S(z)}{z - S(z)\, \Phi(z)} \quad , \tag{29}$$

where $\underline{\mathbf{X}}_t(z) = (X_{1,t}(z), X_{2,t}(z), \ldots)$ and the function $S(z)$ is defined as

$$S(z) \triangleq \prod_{t=1}^{T} S_t(L_t(P_t(z))) \quad . \tag{30}$$

The probability p_0 in (29) can be calculated from the normalization condition $Q(\underline{\mathbf{X}}_1(z), \ldots, \underline{\mathbf{X}}_T(z), z)|_{z=1} = 1$. By using de l'Hôpital's rule, we obtain

$$p_0 = 1 - \rho \quad , \tag{31}$$

where ρ is the load of the system.

In order to obtain the mean buffer content, we calculate the first derivative of (29) with respect to z in the point $z = 1$. This leads to

$$\sum_{t=1}^{T} \sum_{n=1}^{\infty} E[a_n(t)]\; X'_{n,t}(1) + E[u] = \frac{\mathrm{d}}{\mathrm{d}z} \left\{ \frac{\sigma\,(z-1)\, p_0\, S(z)}{z - S(z)\, \Phi(z)} \right\} \Bigg|_{z=1} \quad , \tag{32}$$

where $E[a_n(t)]$ is given by (20). With (26)–(28), after some further calculations, we finally find the following explicit expression for the mean buffer content:

$$E[u] = -\sum_{t=1}^{T} \frac{1}{2} S_t'(1) P_t'(1) \left[\sigma_{L,t}^2 - L_t'(1) + L_t'(1)^2\right] + \frac{1}{2\sigma(1-\rho)} \left\{ \rho\sigma(2-\rho\sigma) + \sum_{t=1}^{T} \left(\sigma_{S,t}^2 L_t'(1)^2 P_t'(1)^2 + \sigma_{L,t}^2 S_t'(1) P_t'(1)^2\right) + \sum_{t=1}^{T} \left(\sigma_{P,t}^2 - P_t'(1)\right) S_t'(1) L_t'(1) \right\} , \tag{33}$$

where $\sigma_{L,t}^2$, $\sigma_{S,t}^2$ and $\sigma_{P,t}^2$ are the variances of the session length, the number of new sessions and the session bandwidth respectively, for sessions of type t.

6.2 PGF of the Buffer Content

The PGF $U(z)$ of u is given by $Q(1, \ldots, 1, z)$. Successive applications of (12) then allow to express $U(z)$ in terms of the function $Q(\underline{\mathbf{X}}_1(z), \ldots, \underline{\mathbf{X}}_T(z), z)$, given in (29). As a result, we obtain

$$U(z) = Q(\underline{\mathbf{X}}_1(z), \ldots, \underline{\mathbf{X}}_T(z), z) \left(\prod_{j=1}^{\infty} \frac{\Phi(z)}{z} g_j(z) \right) + \sigma(z-1) p_0 \sum_{k=1}^{\infty} \frac{1}{\Phi(z)} \left(\prod_{j=1}^{k} \frac{\Phi(z)}{z} g_j(z) \right) , \tag{34}$$

where we have used the property that

$$\lim_{j\to\infty} C_{n,t}(P_t(z) C_{n+1,t}(\ldots P_t(z) C_{j,t}(P_t(z)) \ldots)) = X_{n,t}(z) , \tag{35}$$

$n \geq 1$, $1 \leq t \leq T$, as can be shown from (5), (6) and (26). The function $g_j(z)$ in (34) is defined as

$$g_j(z) \triangleq \prod_{t=1}^{T} S_t(P_t(z) C_{1,t}(P_t(z) C_{2,t}(\ldots P_t(z) C_{j-1,t}(P_t(z)) \ldots))) , \tag{36}$$

and can be further calculated with (5) and (6) as

$$g_j(z) = \prod_{t=1}^{T} S_t\left(P_t(z)^j + \sum_{i=1}^{j-1} \ell_t(i) \left(P_t(z)^i - P_t(z)^j \right) \right) . \tag{37}$$

Combination of (29) and (34) then leads to the following explicit expression for the PGF $U(z)$:

$$U(z) = \frac{\sigma(z-1) p_0 H(z)}{z - S(z)\Phi(z)} , \tag{38}$$

where $H(z)$ is given by

$$H(z) = S(z)\left(\prod_{j=1}^{\infty} \frac{\Phi(z)}{z}\, g_j(z)\right) + [z - S(z)\,\Phi(z)] \sum_{k=1}^{\infty} \frac{1}{\Phi(z)} \left(\prod_{j=1}^{k} \frac{\Phi(z)}{z}\, g_j(z)\right). \tag{39}$$

6.3 Tail Distribution of the Buffer Content

In order to derive an expression for the tail distribution of the buffer content, we will use an approximation technique described in [4]. Specifically, from the inversion formula for z-transforms, it follows that the PMF $\text{Prob}[u = i]$ of u can be expressed as a weighted sum of negative ith powers of the poles of $U(z)$. As the modulus of all these poles is larger than 1, since $U(z)$ is a PGF, it is clear that for large values of i, $\text{Prob}[u = i]$ is dominated by the contribution of the pole of $U(z)$ having the smallest modulus. Let z_0 denote this dominant pole of $U(z)$. The pole z_0 must necessarily be real and positive in order to ensure that the tail distribution is nonnegative anywhere. From (38), it follows that z_0 is a real root of $z - S(z)\,\Phi(z) = 0$. The PMF $\text{Prob}[u = i]$ can then be approximated by the following geometric form:

$$\text{Prob}[u = i] \approx -\frac{\theta_0}{z_0}\left(\frac{1}{z_0}\right)^i , \tag{40}$$

for i sufficiently large, where the constant θ_0 is the residue of $U(z)$ in the point $z = z_0$. This residue can be calculated from (30), (38) and (39) as

$$\begin{aligned}\theta_0 &= \lim_{z \to z_0} (z - z_0)\, U(z) = \frac{\sigma\,(z_0 - 1)\, p_0\, \Phi(z_0)\, H(z_0)}{\sigma - S'(z_0)\,\Phi(z_0)^2} \\ &= \frac{\sigma\, z_0\,(z_0 - 1)\, p_0 \left(\prod_{j=1}^{\infty} \frac{\Phi(z_0)}{z_0}\, g_j(z_0)\right)}{\sigma - \sum_{t=1}^{T} \frac{S_t'(L_t(P_t(z_0)))\, L_t'(P_t(z_0))\, P_t'(z_0)\, z_0\, \Phi(z_0)}{S_t(L_t(P_t(z_0)))}} .\end{aligned} \tag{41}$$

Notice the infinite product in the above expression for θ_0. We know however from (37) that $\lim_{j \to \infty} g_j(z) = S(z)$. Due to the definition of z_0, we moreover have that $S(z_0)\,\Phi(z_0) = z_0$. Therefore, we see that

$$\lim_{j \to \infty} \frac{\Phi(z_0)}{z_0}\, g_j(z_0) = 1 , \tag{42}$$

i.e., the factors of the infinite product in (41) go to 1, as j goes to infinity. Hence, we can calculate the residue θ_0 numerically up to any desired precision by taking the product over a sufficiently large number of factors.

A quantity of considerable interest is the probability that the buffer content exceeds a certain threshold U. Indeed, this quantity is often used to approximate

the packet loss ratio, i.e., the fraction of the arriving packets that is lost upon arrival because of buffer overflow, in a buffer model with a finite storage capacity (for U waiting packets), see e.g. [17]. From (40), we get

$$\text{Prob}[u > U] \approx -\frac{\theta_0}{z_0 - 1} \left(\frac{1}{z_0}\right)^{U+1} , \quad \text{for large } U \ . \tag{43}$$

7 Packet Delay

In this section, we assume a FCFS queueing discipline for packets. We define the delay of a packet as the time interval (expressed in slots) between the end of the packet's arrival slot and the end of the slot during which the packet is transmitted.

In [9], it has been shown that for any discrete-time single-server infinite-capacity queueing system with an FCFS queueing discipline and geometrically distributed packet transmission times (with parameter $1 - \sigma$), regardless of the nature of the arrival process, the following relationship exists between the PGF $D(z)$ of the delay d of an arbitrary packet that arrives in the buffer during a slot in the steady state and the PGF $U(z)$ of the buffer content u after an arbitrary slot in the steady state:

$$D(z) = \frac{U(B(z)) - p_0}{\rho} \ , \tag{44}$$

where $B(z) = \frac{\sigma z}{1-(1-\sigma) z}$ is the PGF of the geometric transmission times.

Since the effective packet transmission times in our model have a geometric distribution, the above relationship is also valid here. It allows us to express the mean value and the tail distribution of the packet delay in terms of the previously derived mean value and tail distribution of the buffer content. In particular, the mean packet delay follows from (44) as

$$E[d] = D'(1) = \frac{U'(1)}{\rho\, \sigma} = \frac{E[u]}{E[m]} \ , \tag{45}$$

in accordance with Little's theorem. For i sufficiently large, the PMF of the packet delay can be approximated as

$$\text{Prob}[d = i] \approx -\frac{\theta_D}{z_D} \left(\frac{1}{z_D}\right)^{i} \ . \tag{46}$$

From (44), it follows that the dominant pole z_D of $D(z)$ is related to the dominant pole z_0 of $U(z)$ as $z_0 = B(z_D)$, or equivalently,

$$z_D = \frac{z_0}{\sigma + (1 - \sigma)\, z_0} \ . \tag{47}$$

The residue θ_D of $D(z)$ in the point $z = z_D$ can be calculated from (44) as

$$\theta_D = \lim_{z \to z_D} (z - z_D)\, D(z) = \frac{\theta_0}{\rho\, B'(z_D)} = \frac{\theta_0\, z_D\, (z_D - 1)}{\rho\, z_0\, (z_0 - 1)} \ . \tag{48}$$

8 Numerical Results and Discussion

In Fig. 3, we assume a single session type ($T = 1$) and show the mean buffer content $E[u]$ as a function of the load ρ. The mean length of the sessions is equal to 100 slots for all of the shown curves, but the distribution of the session lengths is different for each curve. That is, the session lengths respectively are constant, uniform between 1 and 201, negative binomial with two stages, geometric and mixed geometric. For the latter, the mixed geometric distribution of the session lengths has two weighted parallel phases, i.e., the PGF and the mean value are given by

$$L_1(z) = w\,\frac{\gamma_{1,1}\,z}{1-(1-\gamma_{1,1})\,z} + (1-w)\,\frac{\gamma_{1,2}\,z}{1-(1-\gamma_{1,2})\,z} \quad ; \tag{49}$$

$$L_1'(1) = w\,\frac{1}{\gamma_{1,1}} + (1-w)\,\frac{1}{\gamma_{1,2}} \quad . \tag{50}$$

The mean $1/\gamma_{1,1}$ of the first phase is taken to be 50 slots, while the second phase has a mean of 200 slots. The weight w is chosen in order to ensure that $L_1'(1) = 100$. For all curves, the bandwidth of the sessions is fixed at 2 packets per slot, i.e., $P_1(z) = z^2$ and in each slot a new session starts with probability $1/4000$. As in all further examples, the load is increased on the horizontal axis by increasing the mean effective transmission time $1/\sigma$ of the packets. The plot clearly illustrates that the mean buffer content $E[u]$ depends not only on the first moment $L_1'(1)$ of the session-length distribution but on the second-order moment as well. Specifically, (33) predicts a linear impact on the mean buffer content of the variance $\sigma_{L,1}^2$ of the session lengths, which for the considered session-length distributions is 0, 3300, 4999.5, 9900 and 14900 respectively.

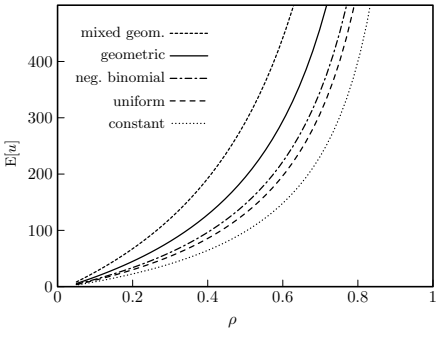


Fig. 3. Mean buffer content as a function of the load ρ in the homogeneous case ($T = 1$) for different session-length distributions $L_1(z)$ with a mean of 100 slots. Bandwidth and session starts have PGF $P_1(z) = z^2$ and $S_1(z) = \frac{3999+z}{4000}$.

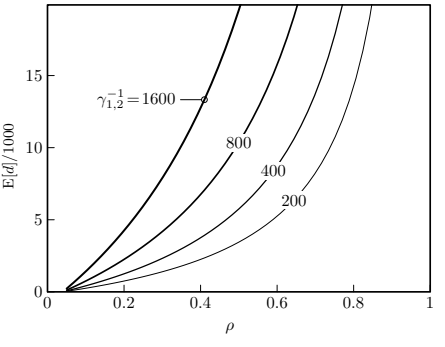


Fig. 4. Mean packet delay as a function of the load ρ in the homogeneous case ($T = 1$) with $P_1(z) = \frac{5}{6}z\,(1-\frac{z}{6})^{-1}$ and $S_1(z) = \frac{2399+z}{2400}$. The session lengths are mixed geometric with mean 100 and first phase mean $1/\gamma_{1,1} = 50$.

This effect is illustrated further in Fig. 4, where the mean packet delay $E[d]$ is shown as a function of the load ρ. Again, the distributions of the session starts and the session bandwidth are the same for all curves, as well as the mean session length $L_1'(1)$ which is 100 slots. The distribution $L_1(z)$ is chosen to be mixed geometric of the form (49) with the first phase mean equal to 50 slots. The plot shows the impact on $E[d]$ if the second phase mean of the session-length distribution is increased, i.e., $1/\gamma_{1,2} = 200$, 400, 800, 1600. For the same configuration and $\rho = 0.8$, the tail distribution of the buffer content $\text{Prob}[u = i]$ is shown in Fig. 5, together with the corresponding mixed geometric distributions of the session lengths.

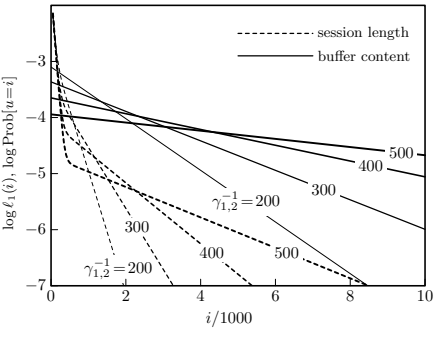


Fig. 5. Logarithmic plot of the mixed geometric session-length distribution and the corresponding tail distribution of the buffer content for load $\rho = 0.8$

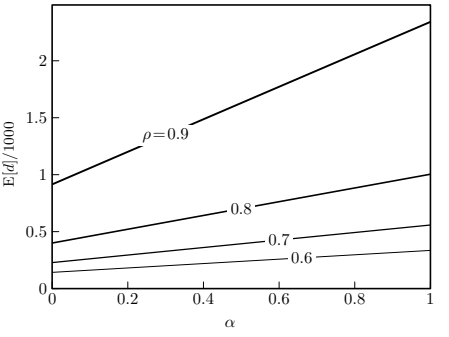


Fig. 6. Mean packet delay as a function of the session mix α. On the left, all sessions are of type 1, while all sessions are of type 2 on the right.

In Fig. 6, we assume heterogeneous traffic with two types of sessions, i.e., $T = 2$. The sessions of type 1 have a constant length of 25 slots, shifted geometric bandwidth $P_1(z) = \frac{z}{2-z}$ with a mean of 2 packets per slot and a Bernoulli session-start distribution with mean $\frac{1-\alpha}{200}$. The sessions of type 2 have a length that is uniformly distributed between 1 and 201, a bandwidth of exactly one packet per slot and a Poisson start distribution with mean $\frac{\alpha}{400}$. The session mix α ($0 \le \alpha \le 1$) indicates the fraction of the load due to sessions of type 1. If $\alpha = 0$, all arrivals are of type 1, while if $\alpha = 1$, there are only sessions of type 2. In Fig. 6, the mean packet delay is shown for load $\rho = 0.6$, 0.7, 0.8, 0.9. We observe a linear dependence of $E[d]$ (and thus also $E[u]$) on the session mix, which is predicted by (33).

In Fig. 7, we consider sessions of T different types and show $E[d]$ as a function of T in case the load is $\rho = 0.6$, 0.7, 0.8, 0.9. For a certain number of types T, the sessions of type t ($1 \le t \le T$) receive an equal share ρ/T of the total load. The sessions of all types have mean length $L_t'(1) = 100$, shifted geometric bandwidth with a mean of $P_t'(1) = 2$ packets per slot and a Bernoulli session-start distribution with mean $\frac{\rho\,\sigma}{100 \cdot 2 \cdot T}$. Also, the session-length distribution for all

types is mixed geometric of the form (49), with first phase mean $1/\gamma_{t,1} = 50$ slots. The tail of the session length however is chosen to be larger for higher types: the second phase mean of type t is $1/\gamma_{t,2} = 100 + 2^t$. Again, we observe a clear impact of the variance of the session lengths on the performance of the system. For $T = 5$, 10 and load $\rho = 0.8$, the tail distributions of the buffer content $\text{Prob}[u = i]$ and the packet delay $\text{Prob}[d = i]$ are shown in Fig. 8.

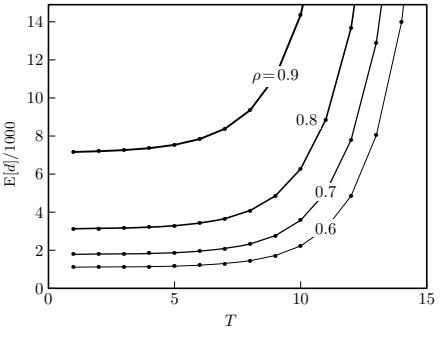


Fig. 7. Mean packet delay as a function of the number of session types T. For type t, the session-length distribution is mixed geometric with mean 100. The phase means are $1/\gamma_{t,1} = 50$ and $1/\gamma_{t,2} = 100 + 2^t$.

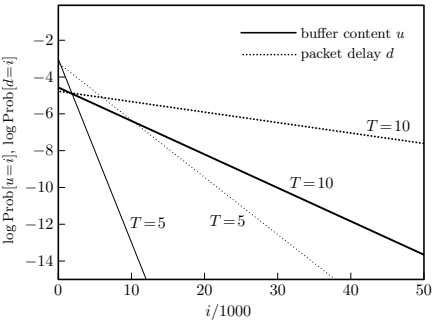


Fig. 8. Logarithmic plot of the tail distributions of the buffer content and the packet delay for $T = 5, 10$ heterogeneous session types and $\rho = 0.8$

9 Conclusions

We have presented an analytical technique for the performance evaluation of a buffer with session-based arrival streams. Differently from previous work, there are T session types and for each type, the session lengths may have a general distribution. Expressions have been obtained for the PGFs, the mean values and the tail distributions of the buffer content and the packet delay. By means of some numerical examples, the impact of the session-based packet arrival process on the performance has been investigated. The results indicate that the buffer behavior strongly depends on the session-length characteristics.

As future work, we plan to study the delay of a session. Also, we intend to apply the model with general session-based arrivals to evaluate the performance of a web server by fitting the model parameters to traces of web traffic.

References

1. Altman, E., Jeanmarie, A.: The distribution of delays of dispersed messages in an M/M/1 queue. In: 14th IEEE Conference on Computer Communications, INFOCOM 1995, Boston (1995)

2. Arlitt, M., Williamson, C.: Internet Web Servers: Workload Characterization and Performance Implications. IEEE ACM Transactions on Networking 5, 631–645 (1997)
3. Bruneel, H.: Packet Delay and Queue Length for Statistical Multiplexers with Low-Speed Access Lines. Computer Networks and ISDN Systems 25, 1267–1277 (1993)
4. Bruneel, H., Kim, B.G.: Discrete-Time Models for Communication Systems Including ATM. Kluwer Academic Publishers, Boston (1993)
5. Choi, B.D., Choi, D.I., Lee, Y., Sung, D.K.: Priority Queueing System with Fixed-Length Packet-Train Arrivals. IEE Proceedings-Communications 145, 331–336 (1998)
6. Cidon, I., Khamisy, A., Sidi, M.: Delay, Jitter and Threshold Crossing in ATM Systems with Dispersed Messages. Performance Evaluation 29, 85–104 (1997)
7. Daigle, J.: Message Delays at Packet-Switching Nodes Serving Multiple Classes. IEEE Transactions on Communications 38, 447–455 (1990)
8. Elsayed, K., Perros, H.: The Superposition of Discrete-Time Markov Renewal Processes with an Application to Statistical Multiplexing of Bursty Traffic Sources. Applied Mathematics and Computation 115, 43–62 (2000)
9. Gao, P., Wittevrongel, S., Bruneel, H.: Delay against System Contents in Discrete-Time G/Geom/c Queue. Electronics Letters 39, 1290–1292 (2003)
10. Hoflack, L., De Vuyst, S., Wittevrongel, S., Bruneel, H.: System Content and Packet Delay in Discrete-Time Queues with Session-Based Arrivals. In: 5th International Conference on Information Technology, ITNG 2008, Las Vegas (2008)
11. Hoflack, L., De Vuyst, S., Wittevrongel, S., Bruneel, H.: Modeling Web Server Traffic with Session-Based Arrival Streams. In: Al-Begain, K., Heindl, A., Telek, M. (eds.) ASMTA 2008. LNCS, vol. 5055, pp. 47–60. Springer, Heidelberg (2008)
12. Kamoun, F.: Performance Analysis of a Discrete-Time Queuing System with a Correlated Train Arrival Process. Performance Evaluation 63, 315–340 (2006)
13. Takagi, H.: Queueing Analysis – A Foundation of Performance Evaluation. Discrete-Time Systems, vol. 3. North-Holland, Amsterdam (1993)
14. Walraevens, J., Wittevrongel, S., Bruneel, H.: A Discrete-Time Priority Queue with Train Arrivals. Stochastic Models 23, 489–512 (2007)
15. Wittevrongel, S.: Discrete-Time Buffers with Variable-Length Train Arrivals. Electronics Letters 34, 1719–1721 (1998)
16. Wittevrongel, S., Bruneel, H.: Correlation Effects in ATM Queues due to Data Format Conversions. Performance Evaluation 32, 35–56 (1998)
17. Woodside, C., Ho, E.: Engineering Calculation of Overflow Probabilities in Buffers with Markov-Interrupted Service. IEEE Transactions on Communications 35, 1272–1277 (1987)
18. Xiong, Y., Bruneel, H.: Buffer Behaviour of Statistical Multiplexers with Correlated Train Arrivals. International Journal of Electronics and Communications 51, 178–186 (1997)

On the Characterization of Product-Form Multiclass Queueing Models with Probabilistic Disciplines

Simonetta Balsamo and Andrea Marin

Università Ca' Foscari di Venezia
Dipartimento di Informatica
via Torino 155, Venezia

Abstract. Probabilistic queueing disciplines are used for modeling several system behaviors. In particular, under a set of assumptions, it has been proved that if the choice of the customer to serve after a job completion is uniform among the queue population, then the model has a BCMP-like product-form solution. In this paper we address the problem of characterizing the probabilistic queueing disciplines that can be embedded in a BCMP queueing network maintaining the product-form property. We base our result on Muntz's property $M \Rightarrow M$ and prove that the RANDOM is the only non-preemptive, non-priority, probabilistic discipline that fulfils the $M \Rightarrow M$ property with a class independent exponential server. Then we observe that the FCFS and RANDOM discipline share the same product-form conditions and a set of relevant performance indices when embedded in a BCMP queueing network. We use a simulator to explore the similarities of these disciplines in non-product-form contexts, i.e., under various non-Poisson arrival processes.

1 Introduction

Queueing models have a pivotal importance for performance evaluation purposes. They have been widely used to model various types of systems, ranging from computer hardware and software architectures, to telephony systems and communication networks. Informally, a queueing model has a set of resources that serves a finite or infinite set of customers. Customers arrive to the model according to a stochastic process (the arrival process) and then they possibly wait for the service in the queue. The service requires a time that is usually modeled by a random variable with a known distribution. After being served, the customers leave the queueing model. The analysis of queueing models, which is usually based on the definition and solution of the underlying stochastic process, provides a set of performance indices including the steady state distribution of the number of customers in the system and some average performance measures, such as the throughput and the mean response time.

In this paper we focus on multiclass queueing models with probabilistic scheduling disciplines. In the simplest queueing models, we usually assume that

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 204–218, 2009.

all the customers are identical, i.e., they all arrive according to the same arrival process and are served according to the same service time distribution. However, for many practical purposes these limitations are unrealistic. In multiclass queueing models, the customers are clustered into classes, and each class is characterized by an arrival process and a service time distribution. The scheduling discipline may also depend on the customer class (e.g., scheduling with priority). According to the literature, in the following we use *RANDOM* to denote the probabilistic queueing discipline in which every customer in the queue has the same probability to enter in service immediately after a job completion, regardless to its arrival time or class. From the performance modeling viewpoint, note that using general probabilistic disciplines allows us to model systems in which the class of a customer may influence the probability of entering in service, thus modeling a sort of mild priority mechanism. For instance this technique is applied to the analysis of the performance of the *Differentiated Services* architecture (RFC 2475) for Internet, as describe in [10]. Among probabilistic disciplines the RANDOM queueing stations have been used to model several resource contention systems, such as shared bus contention systems, as described in [1]. Multiclass queueing models can be embedded with some restrictions in product-form queueing networks (QNs), maintaining the product-form property of the model. Roughly speaking, a QN is a set of interconnected queueing systems that serve a set of customers. A QN is in product-form if its steady sate distribution can be expressed as product of functions whose arguments depend only on the state of one station. The definition of these functions depend on the structure of the network (e.g., the customer routing among the queueing nodes), on the average arrival and service rates and on the queueing discipline. QNs satisfying the well-known BCMP theorem on product-form [3] are multiclass, open, closed and mixed QNs, with probabilistic routing and consisting of stations with four possible scheduling disciplines: First Come First Served (FCFS), Last Come First Served with Preemptive Resume (LCFSPR), Processor Sharing (PS) and Infinite Servers (IS). The service time distribution is exponential for FCFS nodes and general (Coxian) for the other three types of nodes. For these models, exact and efficient analysis techniques have been defined. In [9] the author proves that all these node types fulfil a property called Markov implies Markov ($M \Rightarrow M$) and that every other node type that fulfils that property can be embedded in a BCMP QN maintaining the product-form property. Informally, the $M \Rightarrow M$ property requires that under independent Poisson arrival processes, possibly with different rates for class (class independent Poisson arrivals), a work-conserving multiclass station exhibits departure processes that are independent and Poisson distributed. The main strength of this property is that it defines a relatively easy way to decide if a station leads to a QN with product-form solution by its analysis in isolation and under Poisson arrivals. Note that it is well-known that in a QN with cycles, the arrival processes to a station may be non-Poisson, therefore $M \Rightarrow M$ allows us to study the behavior of a queueing center in a special case (independent Poisson arrivals) and then derive the product-form property of the QN. Using $M \Rightarrow M$, several extensions

of the BCMP theorem have been proposed, considering various disciplines and other constraints (e.g. [1,7]).

In product-form QNs, FCFS and RANDOM disciplines share the same product-form conditions (i.e. exponential service time with the same rate for all the customer classes, i.e. class independent exponential service time [2]) and steady state queue length distribution under an appropriate aggregation of states.

In this paper we study a class of probabilistic queueing disciplines and we focus on the conditions under which multiclass queueing systems with such disciplines admit product-form solutions. We consider the queuing system in isolation, and its analysis as a building block that can be included in a product-form multiclass QN.

We address the problem of identifying possibly product-form queueing systems having a probabilistic discipline that is not RANDOM. In this paper we prove that, if we consider a general BCMP-like product-form solution, under certain conditions, the only probabilistic discipline that leads to a product-from is the RANDOM. In other words, this is a theoretical negative result that states that the extension of BCMP product-form QNs cannot include nodes with probabilistic disciplines, except for the RANDOM.

In order to discuss the effect of probabilistic disciplines on multiclass queueing systems, we present a performance comparison of two queueing systems with RANDOM and FCFS discipline, respectively, by varying the arrival processes. As a consequence of the equivalence results between FCFS and RANDOM stations under class independent Poisson arrivals, for many practical purposes, in a multiclass product-form QN, a RANDOM station can replace a FCFS station. This can be useful for those networks that although they have a BCMP-like product-form, they do not satisfy the conditions of the well-known algorithms for the exact analysis. For instance, this can happen in case of some load-dependent service rates or in case of nodes that are not strictly BCMP (e.g. Le Boudec's MSCCC [7]). In these cases, simulation may be required even for product-form QNs and using the RANDOM disciplines instead of the FCFS allows for a great simplification of the state of the model. In fact, for RANDOM stations, it is not necessary to represent the arrival order but only the number of customers in the queue for each class (therefore the simulation efficiency is improved).

We address the problem of understanding the different behaviors, in terms of stationary queue length distributions and departure processes, of the FCFS and RANDOM stations under stationary non-Poisson arrival processes. By the results of these experiments, we observe that the two queueing systems show a similar behaviors under low load conditions. Under heavy traffic conditions the FCFS and RANDOM queueing systems still have a close queue length distribution, but they exhibit some significant differences in the departure processes. When the different classes have similar arrival rates, we observe a good approximation of the steady state probabilities independently of the load factor, but the departure processes can still exhibit major differences. As a consequence we can derive the conditions under which RANDOM stations can be used to

approximate FCFS stations in complex models that are not in product-form, simplifying the model state and then improving the simulation efficiency.

The paper is structured as follows. Section 2 introduces the queueing model and the definition of the underlying stochastic process. Section 3 reviews the Markov implies Markov property ($M \Rightarrow M$) and introduces the main result of this paper by proving that the RANDOM queueing discipline is the only probabilistic discipline that fulfils the $M \Rightarrow M$ property in queueing station with class independent exponential service time and without preemption or priority. Section 4 presents a performance comparison between the two queuing systems respectively with FCFS and RANDOM disciplines under various arrival processes, through a set of simulation experiments. Section 5 gives some final remarks.

2 Model Description and Notation

In this paper we consider multiclass queueing centers with probabilistic queueing disciplines, single exponential server and class independent stationary arrivals. We name this class of models PQD (Probabilistic Queueing Discipline). Let R be the number of classes, μ the class independent service rate and λ_r the arrival rate of class r customers, for $r = 1, \ldots, R$. When a customer arrives to the queuing system, its service immediately starts if the station is empty, otherwise the customer waits in the queue. At a job completion, a customer is probabilistically chosen from the queue and is immediately put in service. This choice is such that the probability of choosing a customer is non-zero and independent of its arrival time (but e.g. it may depend on its class, on its class arrival rate, on the number of customers of that class in the queue).

Since we are interested in the analysis of the product-form properties of PQD models, we assume exponential class-independent service rates. Indeed, in case of Coxian distributed service times, the station balance condition must hold [4] and this is more restrictive than $M \Rightarrow M$ (e.g. FCFS does not satisfy station balance although it is in product-form in case of class independent exponential service time). Then, in Section 4 we consider PQD models with various arrival distributions and we compare the RANDOM and the FCFS queueing systems with the same service time distribution, in terms of the steady state distributions of the queue lengths and the departure processes.

We denote the state of the model by $\mathbf{m}^{(r)}$, where $\mathbf{m}$ is a R-dimensional vector and $r = 1, \ldots, R$ denotes the class of the customer in service. If the queue is empty we use the symbol $\mathbf{m}_0^{(\epsilon)} \equiv \mathbf{m}_0$, where ϵ means that there is no customer in service. The s-th component of $\mathbf{m}^{(r)}$ is denoted by m_s and represents the number of class s customers in the queue. For instance state $\mathbf{m}^{(r)} = (m_1, \ldots, m_R)^{(r)}$ represents the station with $1 + \sum_{s=1}^{R} m_s$ customers and the customer being served has class r.

At a job completion we probabilistically choose the next customer to be served. We model this behavior by defining a set of non-negative functions $w_s(\mathbf{m}^{(r)})$ that assign a weight to each class $s = 1, \ldots, R$ for every state $\mathbf{m}^{(r)}$. Function

$w_s(\mathbf{m}^{(r)})$ assumes the value 0 if and only if $m_s = 0$. After the job completion the probability of choosing a customer of class s is then given by the following expression:

$$\frac{w_s(\mathbf{m}^{(r)})}{\sum_{t=1}^{R} w_t(\mathbf{m}^{(r)})} \tag{1}$$

It should be clear that if $w_s(\mathbf{m}^{(r)}) = m_s$ then the model is a $M/M/1/RAND$ station since every customer has the same probability of entering in service after a job completion.

In the following, $|\mathbf{m}^{(r)}|_s$ denotes the total number of class s customers in the station, i.e.:

$$|\mathbf{m}^{(r)}|_s = \begin{cases} m_s + 1 & \text{if } r = s \\ m_s & \text{otherwise} \end{cases} .$$

We define $|\mathbf{m}^{(r)}| = \sum_{s=1}^{R} |\mathbf{m}^{(r)}|_s$, i.e., the total number of customers in the station.

3 The RANDOM Queueing Discipline and the BCMP Product-Form

Since BCMP theorem [3] has been formulated, many authors tried to characterize this class of product-form models in terms of structural conditions (e.g. [4]) or conditions on the underlying continuous time Markov chain (CTMC) (e.g. [6,9]).

In particular, in this paper, we focus on the $M \Rightarrow M$ property introduced in [9]. Informally, it states that a multiclass queueing station that, under class independent Poisson arrivals, exhibits class independent Poisson departures, is in product-form. This means that such a queueing station can be embedded in a BCMP QN maintaining the product-form solution of the whole model. It is worthwhile noting that, in general, the arrival processes to the stations within a product-form QN are not Poisson. Therefore, the relevance of the $M \Rightarrow M$ is due to the possibility of studying the behavior of a station in isolation and with independent Poisson arrivals, and then derive the steady state solution when it is embedded in a BCMP-like QN. This property is defined under very general assumptions, i.e., it requires the station to be work-conserving, without priority, and with single-step transitions (see e.g. [5]). In order to prove that the underlying process of a queueing center fulfils the $M \Rightarrow M$ property it suffices to prove that [9]:

$$\forall s = 1, \ldots, R, \quad \forall \gamma \in \Gamma, \qquad \sum_{\xi \in \mathcal{S}^{s+}(\gamma)} \pi(\xi) q(\xi \to \gamma) = \pi(\gamma) \lambda_s, \tag{2}$$

where: Γ is the set of reachable states, $\pi(\gamma)$ is the stationary probability of state $\gamma \in \Gamma$ and $\mathcal{S}^{s+}(\gamma) = \{\xi \in \Gamma : |\xi|_s = |\gamma|_s + 1\}$, i.e., the set of states with one customer of class s more than state γ, where $|\xi|_s$ denotes the number of class s customers in state ξ, $q(\xi \to \gamma)$ is the transition rate between states ξ and γ and λ_s is class s arrival rate.

We can rewrite condition (2) for the multiclass probabilistic queue PQD introduced in the previous section, as follows:

$$\forall s = 1, \ldots, R, \forall \mathbf{m}^{(r)} \in \Gamma \quad \sum_{\mathbf{m}^{(s)} \in \mathcal{S}^{s+}(\mathbf{m}^{(r)})} \pi(\mathbf{m}^{(s)}) q(\mathbf{m}^{(s)} \to \mathbf{m}^{(r)}) = \pi(\mathbf{m}^{(r)}) \lambda_s, \tag{3}$$

where Γ is the (infinite) set of reachable states.

Note that, since there is a single server, if $\mathbf{m}^{(r)}$ can be reached from $\mathbf{m}^{(s)}$ in one step, and $|\mathbf{m}^{(s)}|_t = |\mathbf{m}^{(r)}|_t + 1$, then it follows that $s = t$. Intuitively, this is due to the fact that if $\mathbf{m}^{(s)}$ has a customer of class t more than $\mathbf{m}^{(r)}$ and the latter state can be reached from the former in one step, then that step corresponds to a class s job completion, hence $t = s$.

The main theoretical result of this paper is given by the following theorem. Informally it states that the RANDOM discipline is the *only* probabilistic discipline that fulfils the $M \Rightarrow M$ property for multiclass queueing centers with single server, without preemption, and with class independent exponential service rates. Note that it is well-know that the multiclass RANDOM queueing discipline fulfils the $M \Rightarrow M$ property [1], however, in this paper we prove that it is also *necessary* for those stations that satisfy the conditions of Theorem 1.

Theorem 1. *The RANDOM queue is the only PQD that fulfils the $M \Rightarrow M$ property.*

The proof that RANDOM $\Rightarrow (M \Rightarrow M)$ is given in [1] and the stationary distribution is derived. Hence, we have to prove that $(M \Rightarrow M) \Rightarrow$ RANDOM. In order to prove the theorem, we introduce some definitions and lemmas. Hereafter, for the sake of simplicity, we just write $\mathbf{m}^{(\cdot)}$ to denote a state when the class of the customer being served is not important.

Definition 1. *Let $\mathbf{m}^{(\cdot)}$,$\mathbf{p}^{(\cdot)} \in \Gamma$. We say that $\mathbf{m}^{(\cdot)} \leq \mathbf{p}^{(\cdot)}$ if:*

$$\forall s = 1, \ldots, R \quad |\mathbf{m}^{(\cdot)}|_s \leq |\mathbf{p}^{(\cdot)}|_s,$$

and we define $\mathrm{dist}(\mathbf{p}^{(\cdot)}, \mathbf{m}^{(\cdot)}) = |\mathbf{p}^{(\cdot)}| - |\mathbf{m}^{(\cdot)}|$.

Relation $\leq$ is a partial order in Γ, and $\mathrm{dist}(\mathbf{p}^{(\cdot)}, \mathbf{m}^{(\cdot)})$ is the number of customers that $\mathbf{p}^{(\cdot)}$ has more than $\mathbf{m}^{(\cdot)}$.

Definition 2 (MM-step and MM-path). *Given states $\mathbf{p}^{(s)}$ we call MM-Step a transition to state $\mathbf{m}^{(r)}$ if $|\mathbf{p}^{(s)}|_s = |\mathbf{m}^{(r)}|_s + 1$. We denote a MM-step by: $\mathbf{p}^{(s)} \xrightarrow{r} \mathbf{m}^{(r)}$ and if $\mathbf{m}^{(r)} = \mathbf{m}_0$ we set $r = \epsilon$. A MM-path is a sequence of MM-steps and is described by a vector whose components are the ordered labels of the arrows as well as the initial state.*

In order to help the intuition let us consider a simple example for state $\mathbf{p}^{(s)} = (2,1)^{(2)}$ of a $R = 2$ classes PQD. A possible MM-Path from $\mathbf{p}^{(s)}$ to $\mathbf{m}_0$ is:

$$(2,1)^{(2)} \xrightarrow{1} (1,1)^{(1)} \xrightarrow{2} (1,0)^{(2)} \xrightarrow{1} (0,0)^{(1)} \xrightarrow{\epsilon} \mathbf{m}_0$$

Hence the MM-path is $\alpha = (2,1)^{(2)}; [1,2,1,\epsilon]$. In the following, Greek letters denote MM-pathes. Note that, in general, given a state $\mathbf{m}$, the number of possible MM-paths to $\mathbf{m}_0$ is given by the multinomial coefficient

$$\binom{\sum_{s=1}^{R} m_s}{m_1, \ldots, m_R}.$$

Definition 3 (Function Ψ). *Let $\mathbf{m}^{(\cdot)}, \mathbf{p}^{(\cdot)} \in \Gamma$, with $\mathbf{m}^{(\cdot)} < \mathbf{p}^{(\cdot)}$, and let α be a MM-path between $\mathbf{p}^{(\cdot)}$ and $\mathbf{m}^{(\cdot)}$. The function Ψ is defined as follows:*

$$\Psi(\alpha) = \begin{cases} 1 & \text{if } \alpha = \mathbf{p}^{(\cdot)}; [] \vee \alpha = \mathbf{p}^{(\cdot)}; [\epsilon] \\ \frac{\sum_{s=1}^{R} w_s(\mathbf{p}^{(\cdot)})}{w_r(\mathbf{p}^{(\cdot)})} \Psi(\beta) & \text{otherwise,} \end{cases} \tag{4}$$

where β is the MM-path α with the first MM-step removed.

Intuitively, function Ψ is the reciprocal of the probability that MM-path α occurs given that there has not been any arrival to the queueing center.

The following lemmas state that if the PQD model fulfils the $M \Rightarrow M$ property, then function Ψ only depends on the initial and the final states of the path, i.e., is independent of the order of the MM-steps.

Lemma 1. *If a PQD satisfies the $M \Rightarrow M$ property, then:*

1. *if $(\mathbf{p}^{(s)} \xrightarrow{r} \mathbf{m}^{(r)})$ then we can write:*

$$\frac{\pi(\mathbf{p}^{(s)})}{\pi(\mathbf{m}^{(r)})} = \frac{\lambda_s}{\mu} \frac{\sum_{t=1}^{R} w_t(\mathbf{m}^{(s)})}{w_r(\mathbf{m}^{(s)})}.$$

2. *if $\mathbf{p}^{(s)} \xrightarrow{\epsilon} \mathbf{m}_0$ then we an write:*

$$\frac{\pi(\mathbf{p}^{(s)})}{\pi(\mathbf{m}_0)} = \frac{\lambda_s}{\mu}.$$

Proof. Condition (3) holds by hypothesis, and we already noted that $\mathcal{S}^{s+}(\mathbf{m}^{(r)}) = \{\mathbf{p}^{(s)}\}$. We know that the transition rate from $\mathbf{p}^{(s)}$ to $\mathbf{m}^{(s)}$ is μ multiplied by the probability of choosing a class r customer to put in service. Then the lemma can be derived by trivial algebra. □

Lemma 2. *A PQD satisfies the $M \Rightarrow M$ property if and only if for all states $\mathbf{p}^{(s)} \in \Gamma$ and for all the MM-pathes α from $\mathbf{p}^{(s)}$ to $\mathbf{m}_0$ we have $\Psi(\alpha) = \psi(\mathbf{p}^{(s)})$. Then the stationary probability of state $\mathbf{p}^{(s)}$ can be expressed as follows:*

$$\pi(\mathbf{p}^{(s)}) = \pi(\mathbf{m}_0) \prod_{t=1}^{R} \left(\frac{\lambda_t}{\mu}\right)^{|\mathbf{p}^{(s)}|_t} \psi(\mathbf{p}^{(s)}). \tag{5}$$

Proof. Suppose that the PQD model satisfies the $M \Rightarrow M$ property, and let α be an arbitrary MM-path from $\mathbf{p}^{(r)}$ to $\mathbf{m}_0$. First, we prove by induction that

$$\pi(\mathbf{p}^{(s)}) = \pi(\mathbf{m}_0) \prod_{t=1}^{R} \left(\frac{\lambda_t}{\mu}\right)^{|\mathbf{p}^{(s)}|_t} \Psi(\alpha). \tag{6}$$

Base case: if $\mathbf{p}^{(s)} \xrightarrow{\epsilon} \mathbf{m}_0$ then Equation (6) is verified by Lemma 1 and Definition 3.

Inductive step: suppose that:

$$\alpha = \mathbf{p}^{(s)} \xrightarrow{r} \underbrace{\mathbf{m}^{(r)} \rightarrow \cdots \rightarrow \mathbf{m}_0}_{\beta},$$

and $r \neq \epsilon$. By induction we know that:

$$\pi(\mathbf{m}^{(r)}) = \pi(\mathbf{m}_0) \prod_{t=1}^{R} \left(\frac{\lambda_t}{\mu}\right)^{|\mathbf{p}^{(s)}|_t} \frac{\mu}{\lambda_s} \Psi(\beta). \tag{7}$$

We can write:

$$\frac{\pi(\mathbf{p}^{(s)})}{\pi(\mathbf{m}_0)} = \frac{\pi(\mathbf{p}^{(s)})}{\pi(\mathbf{m}^{(r)})} \frac{\pi(\mathbf{m}^{(r)})}{\pi(\mathbf{m}_0)},$$

that, by Lemma 1 combined with Equation (7) becomes:

$$\frac{\pi(\mathbf{p}^{(s)})}{\pi(\mathbf{m}_0)} = \frac{\lambda_s}{\mu} \prod_{t=1}^{R} \left(\frac{\lambda_t}{\mu}\right)^{|\mathbf{p}^{(s)}|_t} \frac{\mu}{\lambda_s} \Psi(\beta) \frac{\sum_{t=1}^{R} w_r(\mathbf{p}^{(s)})}{w_r(\mathbf{p}^{(s)})}.$$

Using definition 3 we conclude the proof by induction.

Let us consider two different MM-pathes α_1 and α_2 from $\mathbf{p}^{(s)}$ to $\mathbf{m}_0$. By the uniqueness of π for ergodic CTMCs, the following condition is satisfied:

$$\pi(\mathbf{m}_0) \prod_{t=1}^{R} \left(\frac{\lambda_t}{\mu}\right)^{|\mathbf{p}^{(s)}|_t} \Psi(\alpha_1) = \pi(\mathbf{m}_0) \prod_{t=1}^{R} \left(\frac{\lambda_t}{\mu}\right)^{|\mathbf{p}^{(s)}|_t} \Psi(\alpha_2),$$

that implies $\Psi(\alpha_1) = \Psi(\alpha_2) = \psi(\mathbf{p}^{(s)})$. Equation (5) can be obtained from Equation (6) by substitution of Ψ with ψ.

Vice versa let us assume Equation (5) and prove that the $M \Rightarrow M$ property holds. We need to prove Equation (3) rewritten as follows recalling that the transition rate from $\mathbf{p}^{(s)}$ to $\mathbf{m}^{(r)}$ is $\mu w_r(\mathbf{p}^{(s)}) / \sum_{t=1}^{R} w_t(\mathbf{p}^{(s)})$:

$$\pi(\mathbf{p}^{(s)}) \mu \frac{w_r(\mathbf{p}^{(s)})}{\sum_{t=1}^{R} w_t(\mathbf{p}^{(s)})} = \pi(\mathbf{m}^{(r)}) \lambda_s.$$

This can be easily derived by substituting function π by formula (5), in fact:

$$\prod_{t=1}^{R} \left(\frac{\lambda_t}{\mu}\right)^{|\mathbf{p}^{(s)}|_t} \psi(\mathbf{p}^{(s)}) \mu \frac{w_r(\mathbf{p}^{(s)})}{\sum_{t=1}^{R} w_t(\mathbf{p}^{(s)})} = \prod_{t=1}^{R} \left(\frac{\lambda_t}{\mu}\right)^{|\mathbf{m}^{(r)}|_t} \psi(\mathbf{m}^{(r)}) \lambda_s.$$

Noting that $|\mathbf{p}^{(s)}|_s = |\mathbf{m}^{(r)}|_s + 1$ and $|\mathbf{p}^{(s)}|_t = |\mathbf{m}^{(r)}|_t$ with $t \neq s$, we have:

$$\frac{\lambda_s}{\mu}\prod_{t=1}^{R}\left(\frac{\lambda_t}{\mu}\right)^{|\mathbf{m}^{(r)}|_t}\psi(\mathbf{p}^{(s)})\mu\frac{w_r(\mathbf{p}^{(s)})}{\sum_{t=1}^{R}w_t(\mathbf{p}^{(s)})} = \prod_{t=1}^{R}\left(\frac{\lambda_t}{\mu}\right)^{|\mathbf{m}^{(r)}|_t}\psi(\mathbf{m}^{(r)})\lambda_s.$$

This is true if:

$$\psi(\mathbf{p}^{(s)})\frac{w_r(\mathbf{p}^{(s)})}{\sum_{t=1}^{R}w_t(\mathbf{p}^{(s)})} = \psi(\mathbf{m}^{(r)}).$$

Let us consider a MM-path $\alpha = \mathbf{p}^{(s)} \xrightarrow{r} \mathbf{m}^{(r)} \to \cdots \to \mathbf{m}_0$ and $\beta = \mathbf{m}^{(r)} \to \cdots \to \mathbf{m}_0$, by hypothesis $\Psi(\alpha) = \psi(\mathbf{p}^{(s)})$ and $\Psi(\beta) = \psi(\mathbf{m}^{(r)})$, then we can apply Definition 3 and conclude the proof of the lemma. □

Although Lemma 2 states that a PQD model satisfies the $M \Rightarrow M$ property if and only if for every state $\mathbf{p}^{(\cdot)} \in \Gamma$ there exists a function ψ depending only on $\mathbf{p}^{(\cdot)}$ such that $\psi(\mathbf{p}^{(\cdot)}) = \Psi(\alpha)$ for every MM-path α from $\mathbf{p}^{(\cdot)}$ to $\mathbf{m}_0$, it does not give any information about the definition of ψ. The following Lemma gives the definition for ψ from which we straightforwardly derive the proof of Theorem 1.

Lemma 3. *If a PQD model fulfils the $M \Rightarrow M$ property then we can write, for each $\mathbf{p}^{(\cdot)} \in \Gamma$:*

$$\psi(\mathbf{p}^{(\cdot)}) = \binom{\sum_{s=1}^{R} p_s}{p_1, \ldots, p_R}. \tag{8}$$

Proof. The proof is by induction on $|\mathbf{p}^{(s)}|$.

Let $A(\mathbf{p}^{(\cdot)})$ be the set of the class labels with at least one customer in the queue, defined as follows:

$$A(\mathbf{p}^{(\cdot)}) = \{r : p_r > 0\}, \text{ with } r = 1, \ldots, R$$

Note that if $|A(\mathbf{p}^{(\cdot)})| \leq 1$ then we immediately obtain $\psi(\mathbf{p}^{(\cdot)}) = 1$. Therefore if $|\mathbf{p}^{(\cdot)}| \leq 2$ the lemma is trivially satisfied.

Base case: $|\mathbf{p}^{(\cdot)}| = 3$. In this case there is a customer in service and two customers in the queue. If the latter ones belong to the same class then the base case is verified as there is obviously just one possible choice of the class to put in service. Let us consider the case that the queued customers have two different classes s and t. Then the MM-pathes from $\mathbf{p}^{(\cdot)}$ to $\mathbf{m}_0$ are:

$$\alpha : \mathbf{p}^{(\cdot)} \xrightarrow{s} \mathbf{m}^{(s)} \xrightarrow{t} \mathbf{n}^{(t)} \xrightarrow{\epsilon} \mathbf{m}_0$$
$$\beta : \mathbf{p}^{(\cdot)} \xrightarrow{t} \mathbf{m}^{(t)} \xrightarrow{s} \mathbf{n}^{(s)} \xrightarrow{\epsilon} \mathbf{m}_0,$$

hence:

$$\Psi(\alpha) = \frac{w_s(\mathbf{p}^{(\cdot)}) + w_t(\mathbf{p}^{(\cdot)})}{w_s(\mathbf{p}^{(\cdot)})}$$
$$\Psi(\beta) = \frac{w_s(\mathbf{p}^{(\cdot)}) + w_t(\mathbf{p}^{(\cdot)})}{w_t(\mathbf{p}^{(\cdot)})}$$

By Lemma 2 we have that $\psi(\mathbf{p}^{(\cdot)}) = \Psi(\alpha) = \Psi(\beta)$ because by hypothesis the station fulfils the $M \Rightarrow M$ property. Therefore, we conclude $w_s(\mathbf{p}^{(\cdot)}) = w_t(\mathbf{p}^{(\cdot)})$ and $\psi(\mathbf{p}^{(\cdot)}) = 2$.

Induction step: let us consider a state $\mathbf{p}^{(\cdot)}$ with $|\mathbf{p}^{(\cdot)}| > 3$. If $A(\mathbf{p}^{(\cdot)}) = 1$ then the result immediately follows. Let us assume $|A(\mathbf{p}^{(\cdot)})| > 1$ and let $r \in A(\mathbf{p}^{(\cdot)})$. Let us consider an MM-path α to $\mathbf{m}_0$ such that the first MM-step is $\mathbf{p}^{(\cdot)} \xrightarrow{r} \mathbf{m}^{(r)}$. Since $|\mathbf{m}^{(r)}| < |\mathbf{p}^{(\cdot)}|$, by inductive hypothesis we have that:

$$\psi(\mathbf{m}^{(r)}) = \binom{\sum_{s=1}^{R} m_s}{m_1, \ldots, m_R} = \binom{(\sum_{s=1}^{R} p_s) - 1}{p_1, \ldots, p_r - 1, \ldots, p_R}$$

that gives:

$$\Psi(\alpha) = \frac{\sum_{t \in A(\mathbf{p}^{(\cdot)})} w_t(\mathbf{p}^{(\cdot)})}{w_r(\mathbf{p}^{(\cdot)})} \psi(\mathbf{m}^{(r)}). \tag{9}$$

Let us consider $r' \in A(\mathbf{p}^{(\cdot)})$ with $r' \neq r$. In a similar manner we obtain a MM-path β and:

$$\psi(\mathbf{m}^{(r')}) = \binom{\sum_{s=1}^{R} m_s}{m_1, \ldots, m_R} = \binom{(\sum_{s=1}^{R} p_s) - 1}{p_1, \ldots, p'_r - 1, \ldots, p_R},$$

$$\Psi(\beta) = \frac{\sum_{t \in A(\mathbf{p}^{(\cdot)})} w_t(\mathbf{p}^{(\cdot)})}{w_{r'}(\mathbf{p}^{(\cdot)})} \psi(\mathbf{m}^{(r)}).$$

By Lemma 2 we have that $\Psi(\alpha) = \Psi(\beta)$. Then:

$$\frac{1}{w_r(\mathbf{p}^{(\cdot)}) p_{r'}} = \frac{1}{w_{r'}(\mathbf{p}^{(\cdot)}) p_r}.$$

that can be written as:

$$\frac{w_r(\mathbf{p}^{(\cdot)})}{w_{r'}(\mathbf{p}^{(\cdot)})} = \frac{p_r}{p_{r'}}.$$

Since this relation must hold for every couple $r, r' \in A(\mathbf{p}^{(\cdot)})$ and for each state $\mathbf{p}^{(\cdot)}$ that has at least two different classes of customers in the queue (note that when the queue is empty or all the customers belong to the same class the definition of function w is not really important) we conclude that $w_r(\mathbf{p}^{(\cdot)}) = f(\mathbf{p}^{(\cdot)}) p_r$ for every $r \in A(\mathbf{p}^{(\cdot)}$, where f is a non-negative function. By replacing this expression in Equation (9) we conclude the proof of the lemma. □

Proof (Theorem 1). Using the previous lemmas, the theorem proof is very simple. First we observe that the class r weight function w_r influences the model behavior only if there are two or more customers belonging to different classes in the queue. In this case Lemma 3 states that $w_r(\mathbf{p}^{(\cdot)}) = f(\mathbf{p}^{(\cdot)}) p_r$, with $\mathbf{p}^{(\cdot)} \in \Gamma$ and f a non-zero function. Note that f is does not affect the behavior of the PQD because the definition of the probability of entering in service is given by Expression (1). Therefore, we proved that a necessary condition for a PQD to fulfil the $M \Rightarrow M$ property is that the scheduling discipline is RANDOM. □

4 Comparison of RANDOM with FCFS Queueing Discipline

It is well-known that under class independent Poisson arrivals a queueing system with RANDOM or FCFS queueing disciplines is in product-form if the service time distribution is class independent and exponential. In particular, according to the model description given in Section 2 the steady state solution for the RANDOM discipline is:

$$\pi(\mathbf{m}) = \pi_0 \binom{\sum_{r=1}^{R} m_r}{m_1, \ldots, m_R} \prod_{r=1}^{R} \left(\frac{\lambda_r}{\mu}\right)^{m_r}, \tag{10}$$

where π_0 is the stationary probability of observing the empty station, $\mathbf{m}$ is a vector whose components m_r represent the total (in service or in queue) number of customers of class r in the station. The state of FCFS station must represent the arrival order of the customers therefore a straightforward comparison with the expression given by Formula (10) is not possible. However, if we consider an aggregation of states that just represents the number of customers in the station despite to their arrival order, Equation (10) expresses its steady state solution. Note that this equivalence holds even for multiple exponential servers as proven in [2].

In this section we compare the FCFS queueing system with the RANDOM one by assuming class independent exponential service time and under various arrival processes. We focus our attention on the following indices, that are known to be equal in case of Poisson arrivals:

- the steady state probability of observing a given number of customers of a class,
- the distribution of the interdeparture time for a class of customers in steady state. This characterizes the departure process.

By these experiments, we just aim to present an example to illustrate that in multiclass stations the queueing discipline influences the station performance indices, even if it is work-conserving and without priority, and if the service time distribution is class independent and exponential (i.e. with the memoryless property). From this observation we informally derive the conditions under which a multiclass FCFS station embedded in a large model can be approximated by a RANDOM station maintaining the overall model behavior unchanged.

4.1 Experiments

In order to obtain these estimates, we have built a simulator in Java using the combined multiple recursive generator class by L'Ecuyer [8]. Since we are interested in steady state estimates, we used Welch's procedure to define the warmup period [11], performed a set of independent replications of the simulation and constructed 90 percent confidence intervals. The validation of the model has been done with independent Poisson arrivals comparing the estimates with

the exact results. We have considered uniform, hyperexponential and Erlang interarrival time distributions and different load factors.

For the sake of brevity, in this section we show the results of some experiments obtained with the following conditions:

- $R = 3$ classes of customers.
- the interarrival time distributions are Erlang r.v.s with 20 stages of service, with means $1/\lambda_1 = 1.3$, $1/\lambda_2 = 10.0$ and $1/\lambda_3 = 4.0$
- the service rate is exponentially distributed with mean $1/\mu$.

We vary the value of μ in order to change the station load factor $\rho = (\lambda_1 + \lambda_2 + \lambda_3)/\mu$. Since only the classes with the slowest arrival rates present major differences for the considered performance indices, Figures 1, 2, 3 and 4 illustrate the comparison of the two disciplines for classes 2 and 3 and different values of ρ. For all the experiments we observe that the confidence interval width is always less than 0.001, so they are not drawn in the pictures.

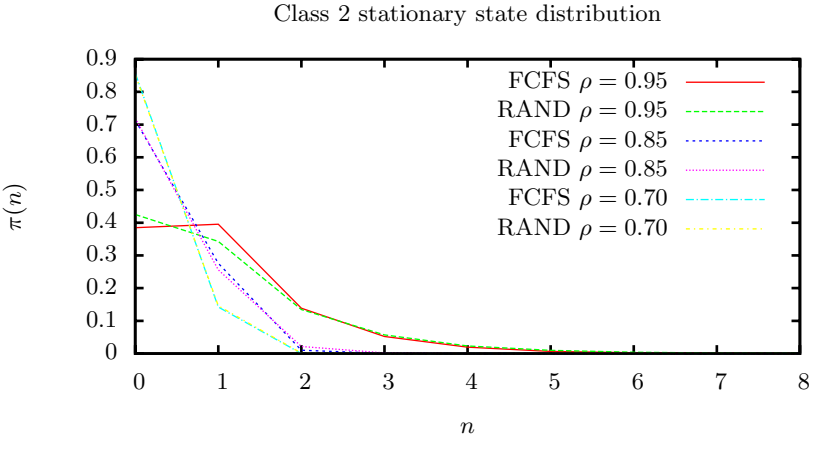


Fig. 1. Class 2 queue population for different values of ρ in steady state

The results clearly show that the queueing disciplines influence the departure processes and the queue population distributions in steady state even if the service time distribution is class independent.

In general we have observed that the FCFS and the RANDOM discipline exhibit similar behaviors in terms of queue population and departure process, if: the class arrival rates are similar, the load factor of the station is low and the interarrival rate distributions can be approximated by independent exponential random variables.

By observing figures 1, 2, 3, 4, it is possible to see the RANDOM and FCFS disciplines exhibit an almost identical behavior, in terms of the considered performance indices, when the load factor $\rho \leq 0.5$. This has an intuitive motivation,

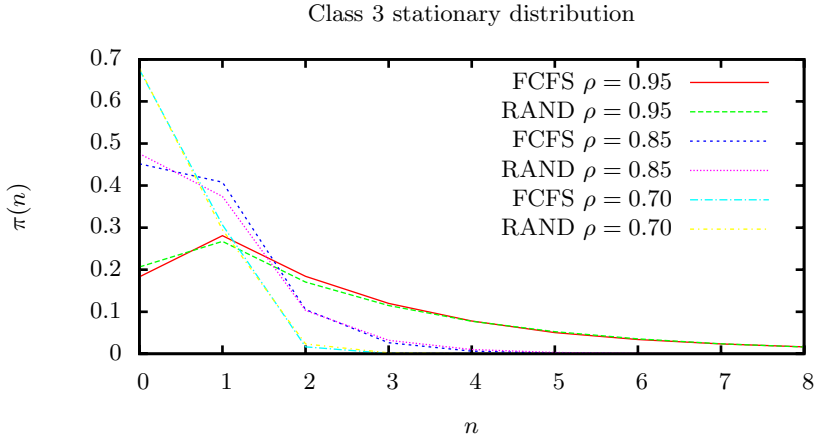


Fig. 2. Class 3 queue population for different values of ρ in steady state

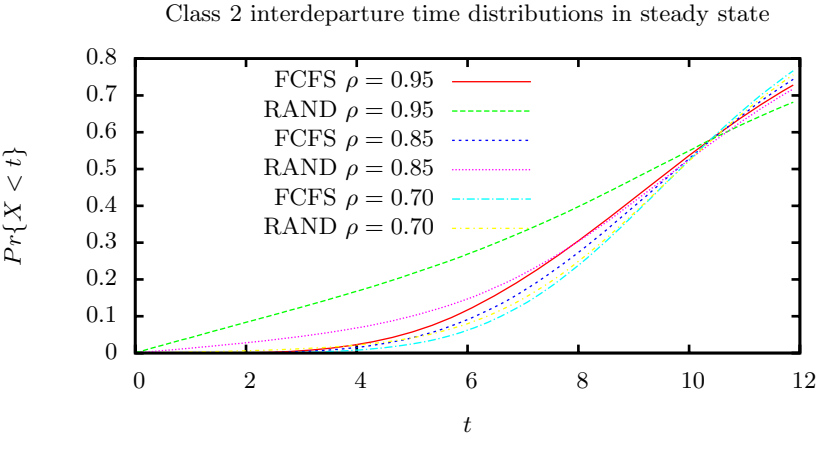


Fig. 3. Class 2 interdeparture time X accumulation function in steady state

i.e., when the traffic is low the probability of observing many customers in the queue is low, therefore the scheduling discipline tends to have a low impact on the considered performance measures. The load factor seems to be a reasonable parameter to analyze if we want to approximate the FCFS stations of a simulated model by RANDOM stations for the sake of improvement of the simulation performance using a more compact state representation.

It is worthwhile pointing out that, for the considered performance indices, the product-form property shows an insensitivity to the scheduling discipline (FCFS or RANDOM) for these multiclass models that is not true in general. Note that this is a peculiarity of multiclass models, while for single class models it is well-known the insensitivity property of work-conserving and non-priority disciplines (e.g. [5]).

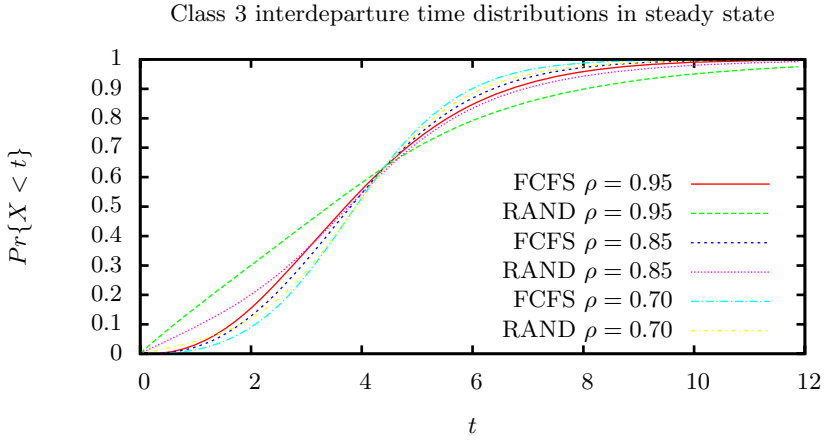


Fig. 4. Class 3 interdeparture time X accumulation function in steady state

5 Conclusions

In this paper we have presented and proved a new theoretical result that characterizes the probabilistic disciplines that fulfil the $M \Rightarrow M$ property, i.e., that can be embedded in a BCMP-like QN maintaining the product-form property of the model. Moreover, we have performed a set of simulation experiments to compare the behaviors of FCFS and RANDOM disciplines under non-Poisson arrival processes. In particular, we have shown the impact of the load-factor on the similarity of the queue population distributions and the departure processes in steady state. In the first part of the paper we have shown that assuming a class independent exponential server the only probabilistic queueing discipline that fulfils the $M \Rightarrow M$ property is the RANDOM. The importance of this result is about the impossibility of defining general non-RANDOM probabilistic disciplines (e.g. [10]) with a class independent exponential server that can be composed in a BCMP-like manner, i.e., that can be studied very efficiently by exact techniques.

In the second part of the paper we have addressed the problem of analyzing the impact on a set of relevant performance indices of the queueing discipline under some non-Poisson class independent arrival processes. The simulation results show that, in case of heavy load, the population distributions and the departure processes of the FCFS and RANDOM models differ. In practice, we explore the possibility of replacing a FCFS station by a RANDOM station (with class independent exponential server) in a non-product-form model. This can be useful because the state representation of the RANDOM station is much more compact than that of FCFS. However, although this leads to exact results in case of product-form QNs, we have observed that it is not generally true. In particular major differences on the steady state population distributions and on the departure processes are observed in case of heavy traffic.

References

1. Afshari, P.V., Bruell, S.C., Kain, R.Y.: Modeling a new technique for accessing shared buses. In: Proc. of the Computer Network Performance Symp., pp. 4–13. ACM Press, New York (1982)
2. Balsamo, S., Marin, A.: On representing multiclass M/M/k queues by generalized stochastic Petri nets. In: Proc. of ECMS/ASMTA-2007 Conf., Prague, Czech Republic, June 4-6, pp. 121–128 (2007)
3. Baskett, F., Chandy, K.M., Muntz, R.R., Palacios, F.G.: Open, closed, and mixed networks of queues with different classes of customers. J. ACM 22(2), 248–260 (1975)
4. Chandy Jr., K.M., Howard, J.H., Towsley, D.F.: Product form and local balance in queueing networks. J. ACM 24(2), 250–263 (1977)
5. Kant, K.: Introduction to Computer System Performance Evaluation. McGraw-Hill, New York (1992)
6. Kelly, F.: Reversibility and stochastic networks. Wiley, New York (1979)
7. Le Boudec, J.Y.: A BCMP extension to multiserver stations with concurrent classes of customers. In: SIGMETRICS 1986/PERFORMANCE 1986: Proc. of the 1986 ACM SIGMETRICS Int. Conf. on Computer performance modelling, measurement and evaluation, pp. 78–91. ACM Press, New York (1986)
8. L'Ecuyer, P.: Good parameters and implementations for combined multiple recursive random number generators. Operations Research 47(1), 159–164 (1999)
9. Muntz, R.R.: Poisson departure processes and queueing networks. Technical Report IBM Research Report RC4145, Yorktown Heights, New York (1972)
10. Tham, C., Yao, Q., Jiang, Y.: A multi-class probabilistic priority scheduling discipline for differentiated services networks. Computer Communications 25(17), 1487–1496 (2002)
11. Welch, P.D.: On the problem of the initial transient in steady-state simulations. Technical report, IBM Watson Research Center, Yorktown Heights, NY (1981)

A Queuing Model for the Non-continuous Frame Assembly Scheme in Finite Buffers

Boris Bellalta

NeTS Research Group - Dpt. of Information and Communication Technologies
Universitat Pompeu Fabra
Passeig de la Circumval-lacio 8, 08003 Barcelona, Spain
boris.bellalta@upf.edu

Abstract. A batch-service finite-buffer queuing model for the non continuous frame assembly scheme is presented. The steady-state probabilities are computed by simply solving the bi-dimensional Markov chain which models the queueing system. However, transitions between states require to know the conditional (as it depends on each state) probability distribution of the on-going batch size, which in turn is computed from the queue departure distribution. Finally, the model is used to evaluate the system performance when the minimum (a) and maximum (b) batch size thresholds are tuned.

1 Introduction

Packet aggregation, from simple packet concatenation to complex data fusion techniques, is used to enhance communication systems. It is able to improve the system performance in terms of throughput, delay and/or energy consumption, by reducing both unnecessary protocol headers and channel contention / transmission attempts. Examples of technologies using aggregation are: Optical Burst Switching [1] or IEEE 802.11n WLANs [2]. Additionally, packet aggregation is also applied on top of a broad set of technologies, such as in mesh networks [3] or in 3G cellular communications [4], in order to enhance the transmission of traffic types which suffer from the inclusion of multiple large headers from the different layers of the protocol stack together with the small portion of useful data (e.g. Voice over Internet Protocol), resulting in a significant waste of transmission resources.

Considering the instant when the batch size is completely determined, two aggregation schemes can be depicted: *i)* Continuous Packet Assembly (CPA), where new incoming packets can be aggregated to the already on-going batch transmission, until the maximum batch size is reached, and *ii)* Non-Continuous Packet Assembly (NCPA), where the batch length is completely determined at the instant it is scheduled, based on the number of packets stored in the queue.

In this paper, only the NCPA scheme is considered. In Figure 1 the transmission queue is shown, where the a and b thresholds are the minimum and maximum allowed batch size, respectively. The queue main features considered are:

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 219–233, 2009.

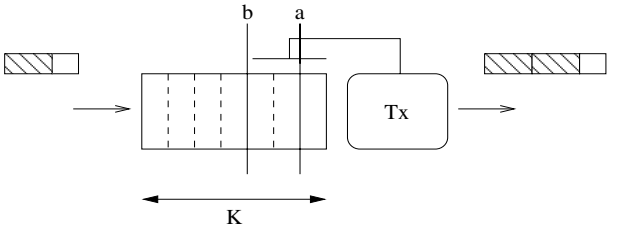


Fig. 1. The NCPA queue

- A packet concatenation strategy is used, where a single header is added to each batch (where a batch is the payload of a group of packets assembled together). The maximum batch size is b packets.
- The transmitter does not store the packets in transmission. They remain in the queue until the batch is completely transmitted.
- Next batch is scheduled as soon as previous transmission has finished and there are at least a packets in the queue. Otherwise, it remains idle.

The proposed model allows to obtain performance metrics such as the packet blocking probability, the average batch size or the average transmission delay. To solve it, a very intuitive relation between the departure $\boldsymbol{\pi^d}$ and the steady-state $\boldsymbol{\pi^s}$ probability distributions is used. This relation is based on the conditional batch length probability distribution $\boldsymbol{\alpha_q}$, which can be computed from the queue departure distribution and is used to determine the transitions inside the set of possible queuing states.

The NCPA queue is based on the queues with bulk-service times [5]. They have received a lot of attention in past years (for example the $GI/M^{[1,b]}/1/K$ [6], the $M/G^{[a,b]}/1/K$ [7] or the $GI/MSC^{[a,b]}/1/K$ [8]), although there are still few works applying these models to real scenarios or communication problems, probably due to its mathematical complexity. Examples are the works from S. Kuppa et al. [9] and Kejie Lu et al. [10] focusing on WLANs performance analysis. Thus, even though the model presented here only works under the assumption of Poisson arrivals and Exponential (batch-length dependant) batch service time distribution. If these assumptions are acceptable, it benefits from two main properties: i) a clear and elegant state-based description (Markov chain) and ii) a striking simplicity which makes it suitable for further enhancements to the packet assembly scheme (e.g. dynamically adapt the b and κ (aggregation factor) parameters to the queuing state) and for its consideration in a joint design with more complex schemes (MAC/PHY protocols).

Once the model is introduced, it is used to show the queue response when the a, b and κ (aggregation factor) parameters are tuned.

2 NCPA Model Description

The NCPA scheme is modeled using a $M/M^{[a,b]}/1_{bd}/K$ (bd:batch dependent) queue with space for K packets, included those in service (there is not a specific

space for them). Packets arrive to the queue according to a Poisson process with rate λ and are served in batches of length l packets, with l taking values between a and b, the minimum and maximum batch size respectively. The batch-service times are exponentially distributed with rate μ_l and depend on the number of packets assembled in each batch. Let q_m be the number of packets in the queue after the departure of the $m-1$ batch. Then, next batch size satisfies the policy $l_m = \beta(q_m)$ where:

$$\beta(q_m) = \begin{cases} a & q_m < a \\ q_m & a \leq q_m < b \\ b & b \leq q_m \end{cases} \tag{1}$$

with the following queue state recursion at departure instants

$$q_m = q_{m-1} - \beta(q_{m-1}) + \min\left(v_{m-1}, K - q_{m-1}\right) \tag{2}$$

where v_{m-1} are the packet arrivals between the $m-2$ and $m-1$ batch departure instants (note that the m batch is scheduled as soon as the $m-1$ batch departs, which leaves q_m packets in the queue).

The average batch service time depends on the aggregation factor κ, the packet length (L, bits), the number of packets assembled together in a single batch (l) and the channel capacity (C, bits/second):

$$\frac{1}{\mu_l} = \frac{L + (l-1) \cdot \kappa \cdot L}{C} \tag{3}$$

The κ parameter must be understood as the proportional part of useful data (payload) in each packet. For example, a packet of length L bits has a payload length equal to $\kappa \cdot L$ bits and a header of length $(1-\kappa) \cdot L$ bits. Thus, the aggregation process consists on, given l individual packets, the extraction of the header from each one and assemble their payload together, adding a single header for the entire batch. This will result in a final batch-length equal to $(1-\kappa)L + l \cdot \kappa \cdot L$ bits.

Note how the κ parameter has a double effect: first, it results in batches of variable size and second, it impacts on the effective traffic load to the queue. Regarding nomenclature, throughout the paper, the traffic load (A) only refers to the load related to the relation between the packet arrival rate and the service time given that no aggregation is done (or equivalently, each batch comprises a single packet) and therefore, $A = \lambda \frac{1}{\mu_1}$.

2.1 Departure Distribution

The limiting departure probability distribution, $\boldsymbol{\pi^d}$, gives the probabilites of having q packets in the queue right after batch departures. It is obtained using the Embedded Markov chain approach, solving the linear system $\mathbf{P}\boldsymbol{\pi^d} = \boldsymbol{\pi^d}$, together with the normalization condition $\boldsymbol{\pi^d}\mathbf{1}^T = 1$. $\mathbf{P}$ is the probability

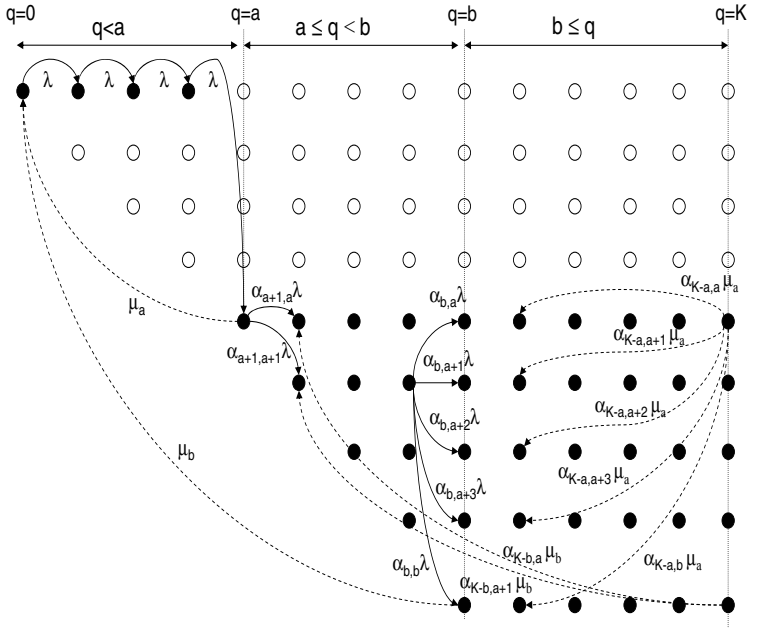


Fig. 2. 2-dimensional Markov chain modelling the NCPA scheme

Table 1. Transition Rates from state $S_{q,l}$, $l\epsilon[a, min(q, b)]$

Next State	Rate	Conditions
Packet arrivals: $q < K$		
$S_{q+1,0}$	λ	$q < a$
$S_{q+1,l'}$	$\alpha_{q+1,l'}\lambda$	$q \geq a$, $l'\epsilon[a, min(q+1, b)]$
Packet departures: $0 < a \leq q \leq K$		
$S_{q-l,0}$	μ_l	$q - l < a$
$S_{q-l,l'}$	$\alpha_{q-l,l'}\mu_l$	$q - l \geq a$, $l'\epsilon[a, min(q-l, b)]$

transition matrix, where the transition probabilities from state $i\epsilon[0, K]$ to state $j\epsilon[0, K]$ are given by:

$$p_{i,j} = \begin{cases} p_{a,j} & i < a, \quad j\epsilon[0, K-a] \\ d_{j+\beta(i)-i,\beta(i)} & i \geq a, \quad j\epsilon[i-\beta(i), K-\beta(i)-1] \\ 1 - \sum_{u=i-\beta(i)}^{K-\beta(i)-1} p_{i,u} & i \geq a, \quad j = K-\beta(i) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $d_{n,l}$ is the probability of n arrivals during the transmission of a batch involving l packets, with

$$d_{n,l} = \int_0^\infty f_v(t) f_l(t) dt = \int_0^\infty e^{-\lambda t} \frac{(\lambda t)^n}{n!} \mu_l e^{-\mu_l t} dt \tag{5}$$

where $f_v(t)$ and $f_l(t)$ are the probability functions of the packet arrivals and the l-batch service time, respectively.

Example. Considering the queue $M/M^{[1,2]}/1_{bd}/5$, the $\boldsymbol{P}$ matrix is given by:

$$\boldsymbol{P} = \begin{bmatrix} d_{0,1} & d_{1,1} & d_{2,1} & d_{3,1} & 1-d_{0,1}-d_{1,1}-d_{2,1}-d_{3,1} & 0 \\ d_{0,1} & d_{1,1} & d_{2,1} & d_{3,1} & 1-d_{0,1}-d_{1,1}-d_{2,1}-d_{3,1} & 0 \\ d_{0,2} & d_{1,2} & d_{2,2} & 1-d_{0,2}-d_{1,2}-d_{2,2} & 0 & 0 \\ 0 & d_{0,2} & d_{1,2} & 1-d_{0,2}-d_{1,2} & 0 & 0 \\ 0 & 0 & d_{0,2} & 1-d_{0,2} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{6}$$

2.2 Steady-State Distribution

Let $\{s(t) : t > 0\}$ be the stochastic process which represents the temporal evolution of the number of packets inside the queue. Its state-space is:

$$\mathcal{S} = \{S_q : \ 0 \leq q \leq K\} \tag{7}$$

and its extended counterpart, including the on-going batch size:

$$\mathcal{S}^e = \{S_{q,l} : \ 0 \leq q \leq K; \ 0 \leq l \leq b\} \tag{8}$$

A sketch of $\mathcal{S}^e$ is shown in Figure 2 (black states), including some representative transitions between states, where the x-axis corresponds to the number of packets in the queue and the y-axis to the size of the current batch in service. The transitions from state $S_{q,l}$ are summarized in Table 1. Let $\boldsymbol{\pi}^{s^e}$ be the limiting steady-state probability vector. It can be obtained by solving the linear system $\boldsymbol{\pi}^{s^e}\mathbf{Q} = \mathbf{0}$, where $\mathbf{Q}$ is the infinitesimal generator of $s^e(t)$, together with the normalization condition, $\boldsymbol{\pi}^{s^e}\mathbf{1}^T = 1$. Note that

$$\pi_q^s = \sum_{j=a}^{min(q,b)} \pi_{q,j}^{s^e}, \qquad q \geq a \tag{9}$$

with $\pi_q^s = \pi_{q,0}^{s^e}$ otherwise. The $(q, 0)$ state means that there are q packets in the queue, although there is not any active transmission.

From Table 1, transition rates between states are partitioned by the conditional batch size distribution $\boldsymbol{\alpha}_q = \{\alpha_{q,l}, a \leq q \leq K, l\epsilon[a, \min(q, b)]\}$, where each $\alpha_{q,l}$ is the probability that given a queuing state q, the system is transmitting a batch of length l.

With respect to the $\boldsymbol{\alpha}_q$ probability distribution, computed in next subsection, it is important to remark that it is the steady-state distribution of the on-going batch length when the queue is in state q, regardless if the queue moves to that state after a new packet arrival or after a departure. This assumption allows the simple model construction, although it introduces some transitions which are physically impossible, such as the one with label $\alpha_{a+1,a+1}$ in Figure 2.

Example. Considering the queue $M/M^{[1,2]}/1_{bd}/5$, the ininitesimal generator $\boldsymbol{Q}$ is given by:

$$\boldsymbol{Q}_\lambda = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\lambda_1 & \alpha_{2,1}\lambda & 0 & 0 & 0 & \alpha_{2,2}\lambda & 0 & 0 & 0 \\ 0 & 0 & -\lambda & \alpha_{3,1}\lambda & 0 & 0 & 0 & \alpha_{3,2}\lambda & 0 & 0 \\ 0 & 0 & 0 & -\lambda & \alpha_{4,1}\lambda & 0 & 0 & 0 & \alpha_{4,2}\lambda & 0 \\ 0 & 0 & 0 & 0 & -\lambda & \alpha_{5,1}\lambda & 0 & 0 & 0 & \alpha_{5,2}\lambda \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{3,1}\lambda & 0 & 0 & -\lambda & \alpha_{3,2}\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_{4,1}\lambda & 0 & 0 & -\lambda & \alpha_{4,2}\lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_{5,1}\lambda & 0 & 0 & -\lambda & \alpha_{5,2}\lambda \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

$$\boldsymbol{Q}_\mu = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu_1 & -\mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu_1 & -\mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_{2,1}\mu_1 & -\mu_1 & 0 & 0 & \alpha_{2,2}\mu_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{3,1}\mu_1 & -\mu_1 & 0 & 0 & \alpha_{3,2}\mu_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_{4,1}\mu_1 & -\mu_1 & 0 & 0 & \alpha_{4,2}\mu_1 & 0 \\ \mu_2 & 0 & 0 & 0 & 0 & 0 & -\mu_2 & 0 & 0 & 0 \\ 0 & \mu_2 & 0 & 0 & 0 & 0 & 0 & -\mu_2 & 0 & 0 \\ 0 & 0 & \alpha_{2,1}\mu_2 & 0 & 0 & 0 & \alpha_{2,2}\mu_2 & 0 & -\mu_2 & 0 \\ 0 & 0 & 0 & \alpha_{3,1}\mu_2 & 0 & 0 & 0 & \alpha_{3,2}\mu_2 & 0 & -\mu_2 \end{bmatrix} \tag{11}$$

where the bidimensional Markov chain, considering only the black states from Figure 2, has been transformed in a single dimensional chain. The infinitesimal generator is $\boldsymbol{Q} = \boldsymbol{Q}_\lambda + \boldsymbol{Q}_\mu$.

2.3 Conditional Batch Size Probability Distribution

Given that there are q packets in the queue, with $a \leq q \leq K$ (otherwise there is no batch transmission), the probability that the length of the on-going batch is l packets, with $a \leq l \leq b$, is:

$$\alpha_{q,l} = \frac{\sum_{i=l}^{q} \mathbf{I}_{\beta(i)=l} \left(p_i^d \cdot p_a(q-i|1/\mu_{\beta(i)}) \right)}{\sum_{j=a}^{q} p_j^d \cdot p_a(q-j|1/\mu_{\beta(j)})} \tag{12}$$

where $\mathbf{I}_{\beta(i)=l}$ is a boolean indicator function which returns 1 if the condition $\beta(i) = l$ is satisfied. Otherwise, it returns 0. Additionally,

$$p_i^d = \begin{cases} \sum_{j=0}^{a} \pi_j^d & i = a \\ \pi_i^d & a < i \leq K \\ 0 & otherwise \end{cases} \tag{13}$$

and $p_a(i|1/\mu_j)$ is the probability of at least i arrivals during the service time with rate $1/\mu_j$.

$$p_a(i|1/\mu_j) = \begin{cases} 1 & i = 0 \\ 1 - \sum_{m=0}^{i-1} d_{m,j} & otherwise \end{cases} \tag{14}$$

Example. Considering the queue $M/M^{[1,2]}/1_{bd}/5$, the $\boldsymbol{\alpha}_4$ probability distribution is given by:

$$\begin{aligned}
\alpha_{4,0} &= 0 \\
\alpha_{4,1} &= \frac{p_1^d \cdot p_a(3|1/\mu_1)}{p_1^d \cdot p_a(3|1/\mu_1) + p_2^d \cdot p_a(2|1/\mu_2) + p_3^d \cdot p_a(1|1/\mu_2) + p_4^d \cdot p_a(0|1/\mu_2)} \\
\alpha_{4,2} &= \frac{p_2^d \cdot p_a(2|1/\mu_2) + p_3^d \cdot p_a(1|1/\mu_2) + p_4^d \cdot p_a(0|1/\mu_2)}{p_1^d \cdot p_a(3|1/\mu_1) + p_2^d \cdot p_a(2|1/\mu_2) + p_3^d \cdot p_a(1|1/\mu_2) + p_4^d \cdot p_a(0|1/\mu_2)}
\end{aligned} \tag{15}$$

Note that $p_i^d = 0$ if $i > K - b$. Expressions from Equation 15 can be simplified accordingly.

2.4 Performance Metrics

Once the $\boldsymbol{\pi}^{\boldsymbol{d}}$ and $\boldsymbol{\pi}^{\boldsymbol{s}^e}$ distributions are obtained, several performance metrics can be computed, such as the packet blocking probability,

$$P_b = \pi_K^s \tag{16}$$

and the probability that the transmitter is empty or non-transmitting:

$$P_{nt} = \sum_{q=0}^{a-1} \pi_q^s \tag{17}$$

Additionally, the average size of the transmitted batches,

$$E[\gamma] = \sum_{i=a}^{K} p_i^d \beta(i) \tag{18}$$

the average number of packets in the queue,

$$E[N] = \sum_{q=0}^{K} q\pi_q^s \tag{19}$$

and the average packet transmission time, obtained from $E[N]$ by applying Little's Law:

$$E[R] = \frac{E[N]}{\lambda(1 - P_b)}. \tag{20}$$

3 Model Validation and NCPA Analysis: Numerical Example

Some numerical and simulation results are provided to validate the accuracy and applicability of the presented queuing model. The $M/M^{[a,b]}/1_{bd}/K$ queue

Table 2. Arrival/Steady-state and Departure distributions for $A = 0.5$ and 1.5 Erlangs

$A = 0.5$ Erlangs	$M/M^{[1,2]}/1_{bd}/5$		$M/M^{[1,3]}/1_{bd}/5$	
State	$\boldsymbol{\pi^d}$	$\boldsymbol{\pi^s}$	$\boldsymbol{\pi^d}$	$\boldsymbol{\pi^s}$
(0)	0.6243 (0.6239)	0.5385 (0.5381)	0.6465 (0.6468)	0.5443 (0.5445)
(1)	0.2359 (0.2363)	0.2473 (0.2477)	0.2279 (0.2276)	0.2454 (0.2452)
(2)	0.0889 (0.0886)	0.1153 (0.1150)	0.0863 (0.0862)	0.1129 (0.1129)
(3)	0.0401 (0.0403)	0.0564 (0.0565)	0.0283 (0.0284)	0.0525 (0.0525)
(4)	0.0106 (0.0106)	0.0254 (0.0255)	0.0107 (0.0108)	0.0253 (0.0253)
(5)	0.0 (0.0)	0.0168 (0.0170)	0.0 (0.0)	0.0193 (0.0194)
$A = 1.5$ Erlangs	$M/M^{[1,2]}/1_{bd}/5$		$M/M^{[1,3]}/1_{bd}/5$	
State	$\boldsymbol{\pi^d}$	$\boldsymbol{\pi^s}$	$\boldsymbol{\pi^d}$	$\boldsymbol{\pi^s}$
(0)	0.2484 (0.2483)	0.1164 (0.1163)	0.3294 (0.3299)	0.1424 (0.1427)
(1)	0.2388 (0.2386)	0.1370 (0.1369)	0.2274 (0.2271)	0.1445 (0.1444)
(2)	0.1739 (0.1739)	0.1387 (0.1386)	0.2423 (0.2422)	0.1592 (0.1592)
(3)	0.2756 (0.2758)	0.1779 (0.1779)	0.1285 (0.1285)	0.1439 (0.1439)
(4)	0.0631 (0.0631)	0.1391 (0.1392)	0.0721 (0.0721)	0.1206 (0.1206)
(5)	0.0 (0.0)	0.2907 (0.2908)	0.0 (0.0)	0.2891 (0.2888)

Table 3. Arrival/Steady-state and Departure distributions for $A = 0.5$ and 1.5 Erlangs

$A = 0.5$ Erlangs	$M/M^{[2,3]}/1_{bd}/5$		$M/M^{[3,3]}/1_{bd}/5$	
State	$\boldsymbol{\pi^d}$	$\boldsymbol{\pi^s}$	$\boldsymbol{\pi^d}$	$\boldsymbol{\pi^s}$
(0)	0.5662 (0.5662)	0.2638 (0.2639)	0.5000 (0.5000)	0.1538 (0.1541)
(1)	0.2452 (0.2457)	0.3781 (0.3784)	0.2500 (0.2503)	0.2307 (0.2313)
(2)	0.1155 (0.1151)	0.1851 (0.1851)	0.2500 (0.2495)	0.3076 (0.3057)
(3)	0.0729 (0.0729)	0.0963 (0.0961)	0.0 (0.0)	0.1538 (0.1561)
(4)	0.0 (0.0)	0.0425 (0.0424)	0.0 (0.0)	0.0769 (0.0754)
(5)	0.0 (0.0)	0.0340 (0.0338)	0.0 (0.0)	0.0769 (0.0772)
$A = 1.5$ Erlangs	$M/M^{[2,3]}/1_{bd}/5$		$M/M^{[3,3]}/1_{bd}/5$	
State	$\boldsymbol{\pi^d}$	$\boldsymbol{\pi^s}$	$\boldsymbol{\pi^d}$	$\boldsymbol{\pi^s}$
(0)	0.2933 (0.2936)	0.0908 (0.0909)	0.2500 (0.2497)	0.0533 (0.0540)
(1)	0.2066 (0.2066)	0.1547 (0.1549)	0.1875 (0.1877)	0.0933 (0.0968)
(2)	0.2508 (0.2505)	0.1609 (0.1609)	0.5625 (0.5624)	0.2133 (0.2441)
(3)	0.2756 (0.2758)	0.1779 (0.1779)	0.0 (0.0)	0.1600 (0.1591)
(4)	0.0631 (0.0631)	0.1391 (0.1392)	0.0 (0.0)	0.1200 (0.1192)
(5)	0.0 (0.0)	0.2907 (0.2908)	0.0 (0.0)	0.3600 (0.3565)

is evaluated using different traffic loads and a and b parameters. A small queue, $K = 5$, is considered in order to be able to show the complete $\boldsymbol{\pi^s}$, $\boldsymbol{\pi^d}$ and $\boldsymbol{\alpha}_q$ values. The channel has a constant capacity equal to $C = 100$ Kbps, $L = 100$ bits (including the header) is the average packet length and $\kappa = 0.5$, resulting in these batch-service times are: $1/\mu_l = [100/C, 150/C, 200/C]$, for $l = 1, \ldots, 3$.

In Table 2 and 3, the departure and steady state distributions obtained by solving the queuing model are compared with the simulation ones (between

Table 4. $\boldsymbol{\alpha}_q$ and $\boldsymbol{\pi}^{se}$ distributions with $A = 1.5$ Erlangs

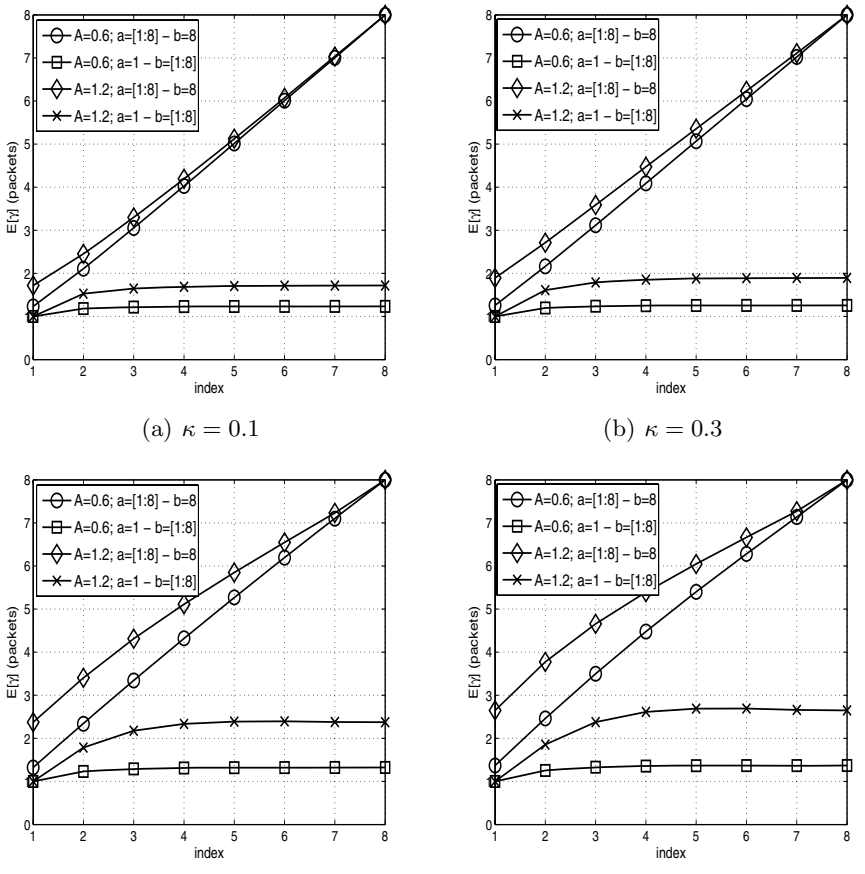
-	$M/M^{[1,2]}/1_{bd}/5$		$M/M^{[1,3]}/1_{bd}/5$		$M/M^{[2,3]}/1_{bd}/5$		$M/M^{[3,3]}/1_{bd}/5$	
State	$\alpha_{q,l}$	$\boldsymbol{\pi}^{se}$	$\alpha_{q,l}$	$\boldsymbol{\pi}^{se}$	$\alpha_{q,l}$	$\boldsymbol{\pi}^{se}$	$\alpha_{q,l}$	$\boldsymbol{\pi}^{se}$
(0,0)	1	0.1164	1	0.1424	1	0.0908	1	0.0533
(1,0)	-	-	-	-	1	0.1547	1	0.0933
(1,1)	1	0.1370	1	0.1445	-	-	-	-
(2,0)	-	-	-	-	-	-	1	0.2133
(2,1)	0.6269	0.0822	0.5795	0.0867	-	-	-	-
(2,2)	0.3730	0.0564	0.4204	0.0725	1	0.1609	-	-
(3,0)	-	-	-	-	-	-	-	-
(3,1)	0.3069	0.0493	0.4035	0.0520	-	-	-	-
(3,2)	0.6930	0.1285	0.3377	0.0502	0.676	0.1114	-	-
(3,3)	-	-	0.2587	0.0416	0.324	0.0578	1	0.1600
(4,0)	-	-	-	-	-	-	-	-
(4,1)	0.2377	0.0296	0.2969	0.0312	-	-	-	-
(4,2)	0.7622	0.1095	0.2860	0.0347	0.6582	0.0771	-	-
(4,3)	-	-	0.4162	0.0546	0.3417	0.0433	1	0.1200
(5,0)	-	-	-	-	-	-	-	-
(5,1)	0.2128	0.0444	0.2586	0.0468	-	-	-	-
(5,2)	0.7871	0.2463	0.2882	0.0782	0.64	0.1735	-	-
(5,3)	-	-	0.4531	0.1640	0.36	0.1301	1	0.3600

Table 5. $E[\gamma]$ (packets) and $E[R]$ (milliseconds) against A Erlangs

-	$M/M^{[1,2]}/1_{bd}/5$		$M/M^{[1,3]}/1_{bd}/5$		$M/M^{[2,3]}/1_{bd}/5$		$M/M^{[3,3]}/1_{bd}/5$	
A	$E[\gamma]$	$E[R]$	$E[\gamma]$	$E[R]$	$E[\gamma]$	$E[R]$	$E[\gamma]$	$E[R]$
0.5	1.139	1.695	1.164	1.687	2.073	2.852	3	4.333
1.0	1.351	2.367	1.429	2.293	2.177	2.756	3	3.592
1.5	1.512	2.780	1.643	2.647	2.249	2.858	3	3.416
2.0	1.621	3.027	1.799	2.855	2.296	2.952	3	3.353
2.5	1.694	3.182	1.911	2.986	2.329	3.003	3	3.325
3.0	1.746	3.286	1.995	3.075	2.353	3.085	3	3.312
3.5	1.783	3.360	2.058	3.139	2.372	3.131	3	3.306

brackets). The simulator has been developed in C programming language using the COST simulation libraries [11]. In Table 4, the steady-state distribution of the expanded state-space, along with the conditional batch distribution are introduced and, finally, in Table 5, the average batch size and the average transmission delay are shown.

At low traffic conditions, comparing the $M/M^{[1,2]}/1_{bd}/5$ and $M/M^{[1,3]}/1_{bd}/5$ queues, the latter shows a higher blocking probability, $P_b = \pi_5^s$. As there is not an specific space for the packets in service and they are only dequeued after the batch is completely transmitted, it results in long periods of time in which there are no packet departures, increasing the probability of losing several consecutive packets. However, this result changes when traffic increases as the



(a) $\kappa = 0.1$ (b) $\kappa = 0.3$

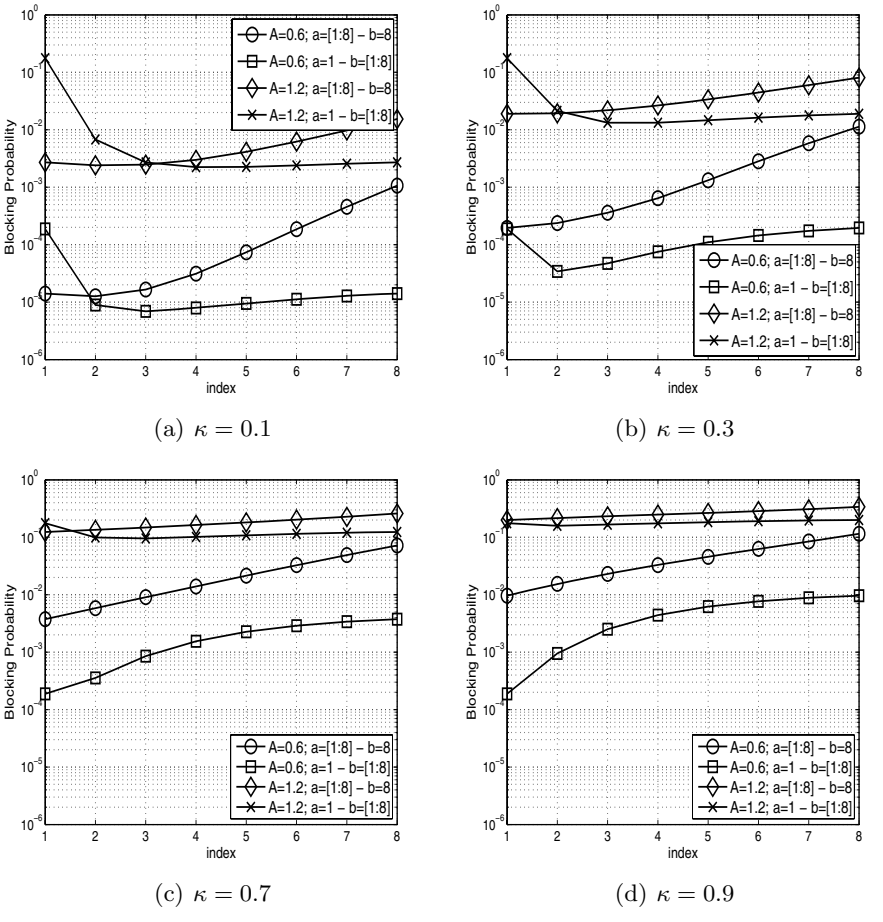
(c) $\kappa = 0.7$ (d) $\kappa = 0.9$

Fig. 3. Average Batch Size

queue $M/M^{[1,3]}/1_{bd}/5$ becomes more efficient. Thus, a higher b value allows to increase the $P_{nt} = \pi_0^s$ probability, as well as the $E[\gamma]$, resulting also in a lower transmission delay, $E[R]$ (Table 5).

On the other hand, the $M/M^{[2,3]}/1_{bd}/5$ and $M/M^{[3,3]}/1_{bd}/5$ increase the a value, which results in higher $P_b = \pi_5^s$ and $P_{nt} = \pi_0^s + \pi_1^s$ and $P_{nt} = \pi_0^s + \pi_1^s + \pi_2^s$ probabilities respectively. At low traffic conditions, the extra delay to schedule a batch is clearly shown, as at least two and three packets are required in each case. This inefficiency is proportionally reduced at high traffic loads, as single packet batches are avoided.

Regarding the values of $\boldsymbol{\alpha}_q$ (Table 4), notice that they also provide a very useful information to understand how the queue performs. For example, notice how increasing b from 2 to 3 in the $M/M^{[1,b]}/1_{bd}/5$ queue results in a higher probability to transmit single packet batches (in Table 4, this is to be in a given state and with a single packet batch active). This, in turn, is caused by



(a) $\kappa = 0.1$ (b) $\kappa = 0.3$

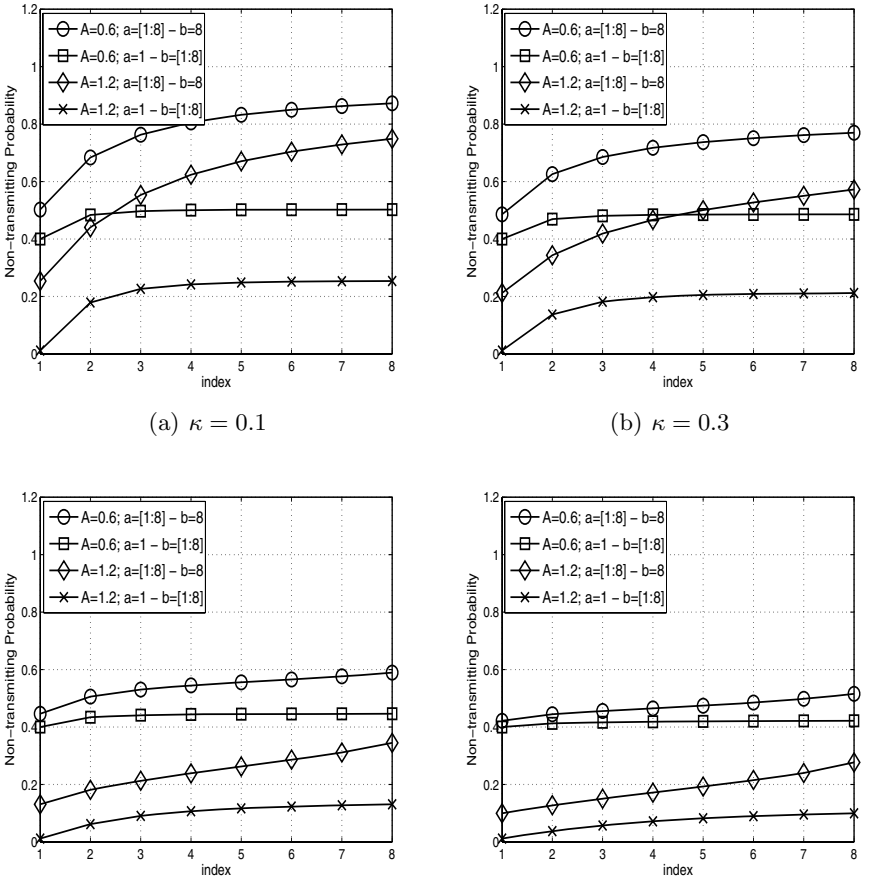
(c) $\kappa = 0.7$ (d) $\kappa = 0.9$

Fig. 4. Blocking Probability

the transmission of these longer batches (3 packets) that now are possible, which left the system at lower states when they depart. Furthermore, increasing a from 1 to 2, it is even worst in terms of increasing the presence of 3 packet batches when there are 4 or 5 packets in the queue.

4 Results: Impact of a and b

This section focuses on the impact of different a and b values, traffic loads ($A = 0.6$ and $A = 1.2$ Erlangs) and κ values (0.1, 0.3, 0.7 and 0.9) on the queue response. The same NCPA parameters used in previous example, with the exception of $K = 15$ and a and b that range from 1 to 8, are considered. Notice that in the Figures, the x-axis values are the index of the set of possible a and b values, which are between brackets in the legend.



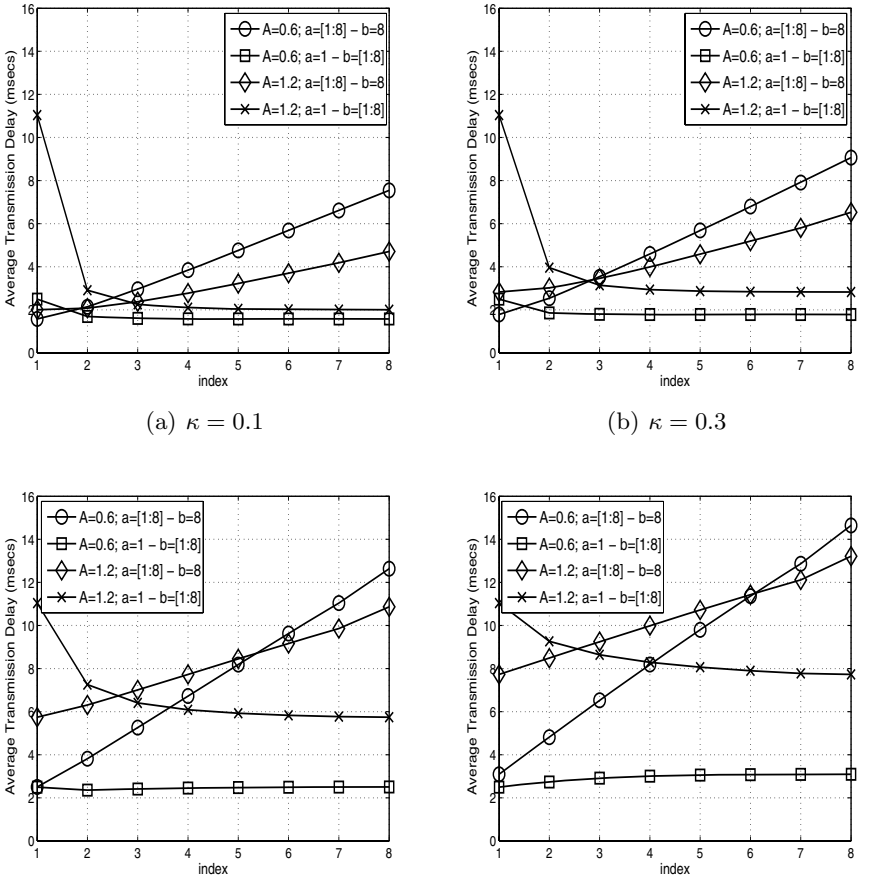
(a) $\kappa = 0.1$ (b) $\kappa = 0.3$

(c) $\kappa = 0.7$ (d) $\kappa = 0.9$

Fig. 5. Non-transmitting probability

In Figure 3, the average batch size ($E[\gamma]$) is shown. As expected, higher traffic loads and κ values result in longer batches (as the batch size is proportional to the number of packets stored in the queue when it is scheduled, that in turn depends on the duration of previous batch). Increasing a, the batch size is constrained to this minimum value. Plus, notice that for a values greater than $K/2$ all scheduled batches will have a size equal to a packets. On the other hand, increasing b results in higher $E[\gamma]$ values until a saturation point is reached, from where $E[\gamma]$ remains approximately constant. Clearly, this saturation point is also proportional to the traffic load and κ and it is also related to the number of packets arriving at the queue during previous batch transmission.

In Figure 4 the blocking probability is plotted. Increasing a results in higher blocking probabilities, except the case with $\kappa = 0.1$. In this case, it is slightly better to wait until a second packet arrives to the queue than to start the batch



(a) $\kappa = 0.1$ (b) $\kappa = 0.3$

(c) $\kappa = 0.7$ (d) $\kappa = 0.9$

Fig. 6. Average Transmission Delay

transmission as soon as possible (note that the low κ makes the aggregation process very efficient). Conversely, when the b parameter is increased, a minimum in the blocking probability appears for some traffic loads and κ values. For higher traffic loads ($A = 1.2$ Erlangs) there is always a minimum (for all κ values). On the contrary, for low traffic loads, it requires low κ values to show that minimum. For example, in the cases with $A = 0.6$ and κ equal to 0.7 and 0.9, always a longer batch results in higher blocking probabilities. These results, in general, are caused by the fact that the packets of a batch are not deallocated until its completion and thus longer batches increase the probability that new arrivals fill the queue.

In terms of the non-transmitting probability (Figure 5), higher b values result always in higher P_{nt}, but not significantly. Thus, although increasing b results in a more efficient use of the channel, the channel utilization still remains high

because of the next batch is scheduled as soon as a single packet ($a = 1$) is ready in the queue (single packet batches are more frequent). Conversely, increasing a this probability boosts substantially as the transmitter remains idle until there are at least a packets. Furthermore, it also causes more efficient aggregations (compared when increasing b), as longer batches are mandatory. Increasing κ, batches also become longer, which increases the channel utilization, thus reducing P_{nt}. A high P_{nt} value means a lower link occupation which in single-user point to point links can result, for example, in lower energy consumption (less transmitted bits or longer sleep periods). However, in random access solutions, it becomes a fundamental parameter as their performance is influenced by the number and persistence of the transmission attempts, which depends on P_{nt} [10]. Then, low P_{nt} values will reduce the collision probability, which can result in higher throughput or lower delay, as well as lower energy consumption.

Finally, the a and b values also show a notably impact in terms of transmission (including queuing) delay (Figure 6). There is a relation (quasi linear) between a, the response delay and the time between packet arrivals. The transmitter remains idle until there are at least a packets in the queue, which is proportional to a and inversely proportional to the traffic load, as $1/\lambda = 1/(A\mu_1)$. On the contrary, increasing b always result in lower queuing delays, except for high κ values (and low traffic loads, Figure 6.d), where the extra delay waiting for the completion of previous batch (which is longer) becomes more relevant than the aggregation gain achieved.

5 Conclusions

A queuing model for the non-continuous frame aggregation scheme in finite buffers has been presented. It is used to provide some directions about the impact of tuning the batch size parameters.

Acknowledgement

This work was partially supported Spanish Government under project TEC2008-06055/TEC. The author would like to specially acknowledge the contribution of the reviewers to improve the final quality of this paper.

References

1. Yao, S., Xue, F., Mukherjee, B., Ben Yoo, S.J., Dixit, S.: Electrical Ingress Buffering and Traffic Aggregation for Optical Packet Switching and Their Effect on TCP-Level Performance in Optical Mesh Networks. IEEE Communications Magazine (September 2002)
2. Xiao, Y.: IEEE 802.11n: enhancements for higher throughput in wireless LANs. IEEE Wireless Communications 12(6), 82–91 (2005)
3. Ganguly, S., Navda, V., Kim, K., Kashyap, A., Niculescu, D., Izmailov, R., Hong, S., Das, S.R.: Performance Optimizations for Deploying VoIP Services in Mesh Networks. IEEE Journal On Selected Areas In Communications 24(11), 2147 (2006)

4. Rittenhouse, G., Zheng, H.: Providing VOIP service in UMTS-HSDPA with frame aggregation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings(ICASSP 2005), vol. 2 (2005)
5. Medhi, J.: Stochastic models in queueing theory. Academic Press, Inc., London (1991)
6. Vijaya Laxmi, P., Gupta, U.C.: On the finite-buffer bulk-service queue with general independent arrivals: $GI/M^{[b]}/1/N$. In: The Third International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine 2006), Waterloo, Ontario, Canada (2006); Operations Research Letters, vol. 25, pp. 241–245 (1999)
7. Chaudhry, M.L., Gupta, U.C.: Modelling and analysis of $M/G^{[a,b]}/1/N$ queue - A simple alternative approach. Queueing Systems 31(1-2), 95–100 (1999)
8. Banik, A.D., Gupta, U.C., Chaudhry, M.L.: Finite-buffer bulk service queue under Markovian service process. In: Proceedings of the 2nd International Conference on Performance evaluation methodologies and tools, Nantes, France (October 2007)
9. Kuppa, S., Dattatreya, G.R.: Modeling and analysis of frame aggregation in unsaturated WLANs with finite buffer stations. In: IEEE International Communications Conference (ICC 2006), Istanbul, Turkey (June 2006)
10. Lu, K., Wang, J., Wu, D., Fang, Y.: Performance of a burst-frame-based CSMA/CA protocol: Analysis and enhancement. Wireless Netw. 15, 87–98 (2007)
11. Chen, G. (G.).: Component Oriented Simulation Toolkit (2004), http://www.cs.rpi.edu/~cheng3/

Equilibrium in Size-Based Scheduling Systems

Sebastien Soudan[1,⋆], Dinil Mon Divakaran[1], Eitan Altman[2], and Pascale Vicat-Blanc Primet[1,⋆⋆]

[1] INRIA / Université de Lyon / ENS Lyon
Tel.: +33-472728037; Fax: +33-472728080
{Sebastien.Soudan,Dinil.Mon.Divakaran,Pascale.Primet}@ens-lyon.fr
[2] INRIA
Tel.: +33-492387786; Fax: +33-492387858
Eitan.Altman@sophia.inria.fr

Abstract. Size-based scheduling is advocated to improve response times of small flows. While researchers continue to explore different ways of giving preferential treatment to small flows without causing starvation to other flows, little focus has been paid to the study of stability of systems that deploy size-based scheduling mechanisms. The question on stability arises from the fact that, users of such a system can exploit the scheduling mechanism to their advantage and split large flows into multiple small flows. Consequently, a large flow in the disguise of small flows, may get the advantage aimed for small flows. As the number of misbehaving users can grow to a large number, an operator would like to learn about the system stability before deploying size-based scheduling mechanism, to ensure that it won't lead to an unstable system. In this paper, we analyse the criteria for the existence of equilibria and reveal the constraints that must be satisfied for the stability of equilibrium points. Our study exposes that, in a two-player game, where the operator strives for a stable system, and users of large flows behave to improve delay, size-based scheduling doesn't achieve the goal of improving response time of small flows.

1 Introduction

Scheduling based on flow size (or flow age) has been gaining importance in the recent times. Researchers have proposed different ways of scheduling based on size, ranging from SRPT (Shortest Remaining Processing Time) to LAS (Least Attained Service) to MLPS (Multi-level Processor Sharing) scheduling mechanisms [1,2,3]. These scheduling strategies differ from the general model for flow scheduling in the Internet. The queues in the Internet nodes, though are served in an FCFS order at packet level, can be modeled using an M/G/1-PS (processor sharing) queue at flow level. The motivation to deviate from this norm,

⋆ Corresponding author.

⋆⋆ This work was done in the framework of the INRIA and Alcatel-Lucent Bell Labs Joint Research Lab on Self Organized Networks.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 234–248, 2009.

and schedule flows based on size, is to give better completion time to small flows. Strictly speaking, the aim has been to improve the conditional mean response time of small flows, at negligible cost to large flows. LAS, for example, always gives highest priority to the flow that has attained the least service. More details on size-based scheduling policies and the advantages they bring, can be found in [4] and [2]. Note that, researchers use *age-based scheduling* to refer to the scheduling schemes that are *blind*, in the sense that, they do not have information about the size of the flow when it arrives, and hence uses its age (the number of bytes/packets already scheduled) to make scheduling decision. Whereas, in this paper, we use the broader phrase *size-based scheduling* to include all the policies that use *age* or *size* to make scheduling decisions.

A user (an end-user or an application) sends a file as a single flow across the Internet. We take this as a normal behaviour. If size-based scheduling is deployed by an operator, there is a clear motivation for one or more users to deviate from the normal behaviour. Indeed, there is an incentive in splitting a flow (possibly large, but more precisely, one that is not small) into multiple small flows to exploit the advantage (say, priority in scheduling) given to small flows to improve the response time. If a considerable number of users deviate from the normal behaviour, then the operator's aim of giving shorter response time to small flows might well be deceived. More importantly, an operator would like to know if such user manipulations would lead to an unstable system behaviour. This poses an important problem in the context of size-based scheduling systems which, to the best of our knowledge, has not been addressed yet. This is the problem we address in this work. In the scenario where users do not misbehave, the stability issue (for network of queues) has been addressed in [5] recently.

The focus of this work is to study the equilibria in size-based scheduling system where users misbehave. We believe this would lead to better understanding of the implication of deploying a size-based scheduling mechanism. More description of the problem is given in Section 2. The model is elaborated in Section 3. The existence of equilibria are studied for two kinds of system behaviours: one in which the service rates are fixed, is studied in Section 4; and the other in which the service rates are varying, is studied in Sections 5 and 6. We summarize our analysis as a game between the operator and users, in Section 7.

2 Problem Statement and Assumptions

We study the problem that arises when an operator deploys a size-based scheduling mechanism. Though there are different ways of scheduling based on size, our focus is on size-based scheduling using two queues. Here, flows are classified based on their sizes. Small flows are sent to one queue, and large flows to another[1]. Each queue is assigned a specific service rate, such that the total service

[1] A flow is called small if its size is less than a threshold, θ. In practice, θ bytes of every large flow also go to the small queue. But, we ignore this to keep the model simple. Besides, this affects neither the analysis nor the results given here.

rate equals the line capacity. The aim of operator in setting such a mechanism is to give to reduce the average response times of small flows.

To formulate the objective of the operator, we assume Poisson flow arrivals. Arrivals and service rates are in units of small flow. λ_x and λ_y are the arrival rates for small and large flows respectively. Each large is F times a small flow. The service rates at small and large queues are ϕ_x and ϕ_y respectively, such that if C denotes the line capacity, $\phi_x + \phi_y = C$. Each queue is served using the PS discipline; hence it is an $M/G/1 - PS$ queue.

We study the existence of equilibria under the scenario where users *cheat* by splitting a large flow into multiple small flows to improve their delay. This is explored in two cases: (i) where the service rates assigned are static, (ii) where the operators exhibits control by dynamically changing the service rates. In the latter case, we explore the existence of interesting equilibria, and state the conditions required for stability, under the assumption that the incentive for players to migrate is to minimize the delay the flow will incur. Note that, by 'players', we consider only the users who migrate.

3 Model Description

The fluid model used in this work is inspired by the one used in [6], where the authors analyse dynamic bandwidth resource allocation and migration between *guaranteed performance* and *best effort* traffic classes.

The two-queues model is depicted in Fig. 1. The queue for small flows is called *small queue* and is referred to as Q_x. The other queue is called the *large queue* which is denoted by Q_y. The number of flows at Q_x is represented by x. At the large queue, this number (in number of small flows) is denoted by y. We assume infinite queues. The service rates, ϕ_x and ϕ_y, are also in number of small flows. They are both assumed to take non-zero values.

The system parameters ϕ_x and ϕ_y are set by the operator. System state is modeled using averaged queue sizes: x and y. Depending on the measured delay values, a user might decide to split a large flow into multiple small flows. Therefore, a fraction of the flows arriving at the large queue might be *migrated* to the small queue. This migration function, which is a result of aggregate user behaviour, is represented as $m(x, y)$. It is linear in $\lambda_y F$ as a result of the integration of individual user that send $\mathrm{d}\lambda_y$ each: $\int m \mathrm{d}\lambda_y = \lambda_y m$. We take m to be a non-negative and continuous function of x and y. m represents the fraction of λ_y which goes to Q_x.

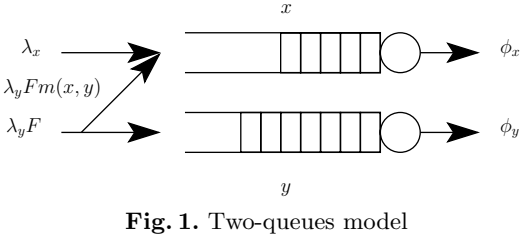


Fig. 1. Two-queues model

$$0 \leq m(x,y) \leq 1 \tag{1}$$

For every large flow that migrates, it adds an overhead of η (e.g. connection establishment cost, slow-start cost). The rate equations can now be written as:

$$\frac{dx}{dt} = \lambda_x - \phi_x + \lambda_y F m(x,y)(1+\eta), \quad x > 0 \tag{2}$$

$$\frac{dy}{dt} = \lambda_y F - \phi_y - \lambda_y F m(x,y), \quad y > 0 \tag{3}$$

The rate equations are different at the borders. For $x = 0$,

$$\left.\frac{dx}{dt}\right|_{x=0} = [\lambda_x - \phi_x + \lambda_y F m(x,y)(1+\eta)]^+ \tag{4}$$

and for $y = 0$,

$$\left.\frac{dy}{dt}\right|_{y=0} = [\lambda_y F - \phi_y - \lambda_y F m(x,y)]^+. \tag{5}$$

4 System Analysis for Static Service Rates

This section details the analysis of a system where the service rates at both the queues are fixed.

Proposition 4.1. *An interior point (x,y) is an equilibrium iff $\phi_x - \lambda_x = \lambda_y F - \phi_y$ and m is such that $m(x,y) = \frac{\phi_x - \lambda_x}{\lambda_y F}$ and $0 \leq m(x,y) \leq 1$.*

Proof (Proof of Prop. 4.1)
Let (x,y) be an interior point. It is an equilibrium if and only if:

$$\begin{cases} \frac{dx}{dt} = 0 \\ \frac{dy}{dt} = 0 \\ 0 \leq m(x,y) \leq 1 \end{cases} \iff \begin{cases} m(x,y) = \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ m(x,y) = \frac{\lambda_y F - \phi_y}{\lambda_y F} \\ 0 \leq m(x,y) \leq 1 \end{cases}$$ □

Remark 4.2. *Existence of interior equilibrium does not only depend on m function but also on the arrival rates and service rates. Meaning that they can only exist in very specific cases.*

Proposition 4.3. $(0,0)$ *is an equilibrium point if and only if:*

$$\begin{cases} m(0,0) \leq \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ \frac{\lambda_y F - \phi_y}{\lambda_y F} \leq m(0,0) \\ 0 \leq m(0,0) \leq 1 \end{cases}$$

Proof (Proof of Prop. 4.3). Using equations (4) and (5), we obtain that $(0,0)$ is an equilibrium point if and only if:

$$\begin{cases} \frac{dx}{dt}\Big|_{x=0} = 0 \\ \frac{dy}{dt}\Big|_{y=0} = 0 \\ 0 \leq m(0,0) \leq 1 \end{cases} \iff \begin{cases} \frac{\lambda_x - \phi_x}{1+\eta} + \lambda_y F m(0,0) \leq 0 \\ \lambda_y F - \phi_y - \lambda_y F m(0,0) \leq 0 \\ 0 \leq m(0,0) \leq 1 \end{cases}$$

$$\iff \begin{cases} m(0,0) \leq \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ \frac{\lambda_y F - \phi_y}{\lambda_y F} \leq m(0,0) \\ 0 \leq m(0,0) \leq 1 \end{cases}$$

□

Proposition 4.4. *$(0,y)$ with $y > 0$ is an equilibrium point if and only if:*

$$\begin{cases} m(0,y) \leq \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ m(0,y) = \frac{\lambda_y F - \phi_y}{\lambda_y F} \\ 0 \leq m(0,y) \leq 1 \end{cases}$$

Proof (Proof of Prop. 4.4). Using equations (4) and (3), we obtain that $(0,y)$ is an equilibrium point if and only if:

$$\begin{cases} \frac{dx}{dt}\Big|_{x=0} = 0 \\ \frac{dy}{dt} = 0 \\ 0 \leq m(0,y) \leq 1 \end{cases} \iff \begin{cases} \frac{\lambda_x - \phi_x}{1+\eta} + \lambda_y F m(0,y) \leq 0 \\ \lambda_y F - \phi_y - \lambda_y F m(0,y) = 0 \\ 0 \leq m(0,y) \leq 1 \end{cases}$$

$$\iff \begin{cases} m(0,y) \leq \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ m(0,y) = \frac{\lambda_y F - \phi_y}{\lambda_y F} \\ 0 \leq m(0,y) \leq 1 \end{cases}$$

□

Proposition 4.5. *$(x,0)$ with $x > 0$ is an equilibrium point if and only if:*

$$\begin{cases} m(x,0) = \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ \frac{\lambda_y F - \phi_y}{\lambda_y F} \leq m(x,0) \\ 0 \leq m(x,0) \leq 1 \end{cases}$$

Proof (Proof of Prop. 4.5). Using equations (2) and (5), we obtain that $(x,0)$ is an equilibrium point if and only if:

$$\begin{cases} \frac{dx}{dt} = 0 \\ \frac{dy}{dt}\Big|_{y=0} = 0 \\ 0 \leq m(0,y) \leq 1 \end{cases} \iff \begin{cases} \frac{\lambda_x - \phi_x}{1+\eta} + \lambda_y F m(x,0) = 0 \\ \lambda_y F - \phi_y - \lambda_y F m(x,0) \leq 0 \\ 0 \leq m(x,0) \leq 1 \end{cases}$$

$$\iff \begin{cases} m(x,0) = \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ \frac{\lambda_y F - \phi_y}{\lambda_y F} \leq m(x,0) \\ 0 \leq m(x,0) \leq 1 \end{cases}$$

□

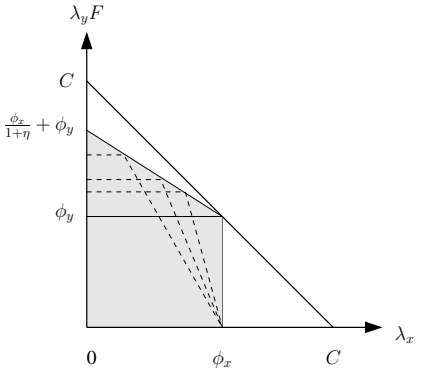


Fig. 2. Existence region of equilibrium $(0, 0)$ under static service rate

4.1 Discussion

The aim of a network operator in deploying such a scheduling mechanism is to give shorter delays to small flows, at negligible cost to large flows. With this in mind, we can now evaluate which among the equilibrium points are interesting and useful (from the perspective of a network operator).

To start with, let us consider the equilibrium point $(0, 0)$. The inequalities of Prop. 4.3 give the shaded region of Fig. 2, where one m can exist to make $(0, 0)$ an equilibrium. This region is dominated by the line $\lambda_x + \lambda_y F = C$, which defines the region where a single queue system would have empty queue equilibrium. Thus, this equilibrium (in the two queue system) is not of great interest for the network operator.

The lines $(x, 0)$ and $(0, y)$ constitute the remaining border point equilibria. $(x, 0)$ is the set of those points where there is queueing in the small queue, but not at the large queue. For this reason, these are not desirable equilibria from operator's point of view. Similarly existence of $(0, y)$ means, there is nothing queueing at Q_x. So, there is incentive for users to migrate to Q_x. Hence $(0, y)$ will not be stable.

As seen in previous section, interior point equilibrium are only possible in limiting cases where the surplus rate at the large queue is exactly equal to the surplus of service of x, with the additional constraint that m transfers exactly this. This situation is too constrained to happen in a real scenario. To introduce more flexibility, the operator can control the service rate. But this requires the use of some observable parameters of the system. In this system, the only observable parameters are x and y as arrival rates λ_x and λ_y are not separable at the queues.

5 Control on ϕ_x Using Parameter x

In this section we study the system when operator controls the service rates using a single parameter. Let f be the control function, and x be the control parameter. In the remaining of this section we use the following definition for $\phi_x(x)$ and $\phi_y(x)$.

Definition 5.1. *$\phi_x(x)$ and $\phi_y(x)$*
$\phi_x(x) = f(x)$
$\phi_y(x) = C - f(x)$

C being the maximum link capacity (or service rate), let:

$$0 < f(x) < C \tag{6}$$

so that the service rate at any queue doesn't vanish.

5.1 Delay Condition

We introduce the delay condition which is satisfied at equilibrium, as the users have no incentive to migrate once the delays at both queues are equal. Let us look the delay a large flow will incur Q_x, if it is split into F small flows. For a service rate of ϕ_x at Q_x, each small flow gets $\frac{\phi_x}{x+F}$ of service. Hence the time to transfer a large flow through Q_x is $T_x = \frac{x+F}{F\phi_x}(1+\eta)$. On the other hand, if the arriving large flow decides to queue at Q_y, the delay experienced will be $T_y = \frac{y+1}{\phi_y}$.

At equilibrium, $T_x = T_y$; thus,

$$\frac{(x+F)(1+\eta)}{F\phi_x} = \frac{y+1}{\phi_y} \tag{7}$$

5.2 Analysis of Equilibrium

For equilibrium to exist, the equations (2) and (3) should be equated to zero.

Proposition 5.2. *If η is zero, no equilibrium will exist unless $C = \lambda_x + \lambda_y F$.*

Proof (Proof of Prop. 5.2).

From the combination (2) + (3) at equilibrium, when η is 0, we get $C = \lambda_x + \lambda_y F$. □

In the remaining, η is taken to be strictly positive.

Using equations (2) and (3) at equilibrium, gives the constraints (8) on f for the existence of such an equilibrium point.

$$f(x_e) = \frac{(1+\eta)(C - \lambda_y F) - \lambda_x}{\eta} \tag{8}$$

There can be multiple such points x_e or no depending on f.

Proposition 5.3. *For a given set of parameters $(\lambda_x, \lambda_y, C, \eta, f, m)$ with $\eta > 0$, the system has inner equilibrium points (x_e, y_e) where:*

$$x_e \in f^{-1}\left(\frac{(1+\eta)(C - \lambda_y F) - \lambda_x}{\eta}\right) \tag{9}$$

and:

$$y_e = \frac{C - f(x_e)}{F f(x_e)}(x_e + F)(1 + \eta) - 1$$

iff:

$$\begin{cases} \lambda_x + \lambda_y F \leq C \\ \lambda_x + \lambda_y F(1+\eta) > C \\ f^{-1}(\frac{(1+\eta)(C-\lambda_y F)-\lambda_x}{\eta}) \neq \emptyset \\ m(x_e, y_e) = \frac{C-(\lambda_x+\lambda_y F)}{\eta \lambda_y F} \end{cases} \tag{10}$$

Proof (Proof of Prop. 5.3)

From the combination, (2) + $(1+\eta)$(3), at equilibrium, we obtain Eq. (8). The system has equilibriums iff there is point x_e satisfying this equation, meaning $f^{-1}(\frac{(1+\eta)(C-\lambda_y F)-\lambda_x}{\eta})$ is not empty ($\eta \neq 0$). From Eq. (7), we get corresponding y_e. Then from Eq. (2) at equilibrium, we have m as defined in Eq. (10).

Due to constraint (1) on m, and constraint (6) on f, we have the existence of this equilibrium iff:

$$\begin{cases} \lambda_x + \lambda_y F \leq C \\ C < \lambda_x + \lambda_y F(1+\eta) \end{cases} \tag{11}$$

Second inequality is strict because of Eq. (6). □

Fig. 3 shows the region of arrival rates where equilibrium can exist, dashed-line is excluded from this.

Corollary 5.4. *If f is strictly monotonic. For every 2-tuple of (λ_x, λ_y) satisfying the line equation: $(1+\eta)(C - \lambda_y F) - \lambda_x = k$ (for a constant k), there is maximum of one equilibrium point.*

Proof. Corollary 5.4

If f is strictly monotonic, there is utmost one pre-image by f^{-1}. As potential equilibria are determined by Eq. (9) (and y_e which only depends on x_e), all points of the line of arrival rates: $(1+\eta)(C - \lambda_y F) - \lambda_x = k$ have the same potential equilibrium. Since $m(x_e, y_e)$ has to satisfy Eq. (10), which gives a different line in λ_x and λ_y, there is at most one equilibrium point (the intersection). □

From the above, it can be observed that, for a monotonic f, there exists utmost one equilibrium point for the whole line of arrival rates. This gives only a few equilibrium points for a wide range of arrival rates. A non-monotonic f will give more equilibrium points. But still, it is not feasible to obtain equilibrium points for all values of (λ_x, λ_y) satisfying the line of arrival rates, as it would require an infinite queue or an infinite variability of f.

Hence, we conclude that control using a function of x alone, is not of any use to the operator.

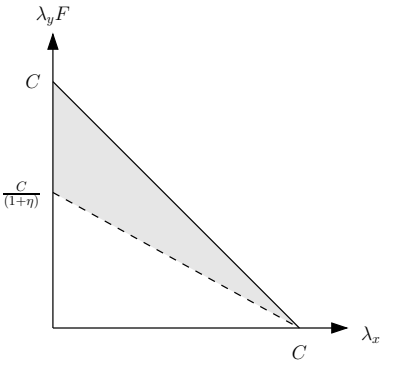


Fig. 3. Interior equilibrium existence region under $\phi_x(x) = f(x)$

6 Control on ϕ_x Using Parameters x and y

As seen in previous section, using only one parameter is not enough to stabilize the system as the control space is too small. We thus use a control function with two parameters: x and y.

Definition 6.1. *$\phi_x(x, y)$ and $\phi_y(x, y)$*

$\phi_x(x, y) = g(x, y)$
$\phi_y(x, y) = C - g(x, y)$

Similar to what have been done with f, if C is the maximum link capacity (or service rate), let:

$$0 < g(x, y) < C \tag{12}$$

Note that, definition of delay equation at equilibrium as given in (7) remains the same and so we directly proceed to the analysis of potential equilibria.

6.1 Analysis of Equilibrium

For equilibrium to exist, the equations (2) and (3) should be equated to zero. Prop. 5.2 still holds in this case as ϕ_x and ϕ_y also sum to C; therefore from equations (2) and (3) we can prove the same. Hence η is also taken strictly positive here.

Similarly to what have been done for Prop.5.3, at equilibrium, using Eq. (2) and (3), we obtain the following constraint on g:

$$g(x_e, y_e) = \frac{(1+\eta)(C - \lambda_y F) - \lambda_x}{\eta} \tag{13}$$

Proposition 6.2. *For a given set of parameters* $(\lambda_x, \lambda_y, C, \eta, g, m)$ *with* $\eta > 0$, *the system has inner equilibrium points* (x_e, y_e) *iff:*

$$\begin{cases} g(x_e, y_e) = \frac{(1+\eta)(C-\lambda_y F)-\lambda_x}{\eta} \\ m(x_e, y_e) = \frac{C-(\lambda_x+\lambda_y F)}{\eta\lambda_y F} \\ \frac{(x+F)(1+\eta)}{F\phi_x} = \frac{y+1}{\phi_y} \\ \lambda_x + \lambda_y F \leq C \\ \lambda_x + \lambda_y F(1+\eta) > C \end{cases} \tag{14}$$

Proof (Proof of Prop. 6.2). Same as Prop. 5.3 except that (8) has been replaced by (13). □

Note that the region of arrival rates where equilibrium points can exist is the same.

We define an *equivalent load* Γ:

Definition 6.3. $\Gamma(\lambda_x, \lambda_y) = \frac{\lambda_x}{1+\eta} + \lambda_y F$.

Definition 6.4. $D(\Gamma)$ *is the set of* (x, y) *satisfying:*

$$y = a(\Gamma)x + b(\Gamma) \tag{15}$$

where

$$a(\Gamma) = \frac{(1+\eta)\Gamma - C}{F(C-\Gamma)}$$

and

$$b(\Gamma) = \frac{(2+\eta)\Gamma - 2C}{C-\Gamma}$$

Proposition 6.5. *For a given setting of arrival rates* (λ_x, λ_y) *satisfying*

$$\Gamma(\lambda_x, \lambda_y) = k \tag{16}$$

and the two inequalities of Prop. 6.2, equilibria (x_e, y_e) *under this load are on* $D(k)$. *Besides, for all the equilibrium points in* $D(k)$, g *satisfies* (13) *and is constant:*

$$g(x_e, y_e) = \frac{1+\eta}{\eta}(C-\Gamma) \tag{17}$$

Proof (Proof of Prop. 6.5). Let (λ_x, λ_y) be a setting of arrival rates satisfying Eq. (16) and the two inequalities of Prop. 6.2.

We first show that $D(k)$ contains all the potential equilibrium points. By replacing g using Eq. (13) in the delay equation (7), we obtain Eq. (15).

All equilibrium points of arrival settings satisfying (16) have the same value of g as Eq. (13) holds and gives Eq. (17) which is constant in $\Gamma(\lambda_x, \lambda_y)$. □

Fig. 4 shows Γ-lines in the $\lambda_x\lambda_y$-plane and their corresponding $D(\Gamma)$-lines in the xy-plane. On each such line in the xy-plane, g is constant and thus gradient is orthogonal. From Eq. (17), we also know $\frac{\partial g}{\partial \Gamma}$ is negative which justifies the orientation of gradient on the figure.

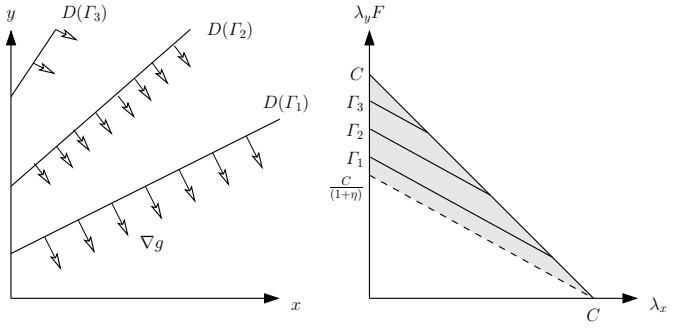


Fig. 4. Interior equilibrium existence region and mapping of $\Gamma(\lambda_x, \lambda_y)$ lines to $D(\Gamma)$, level sets and gradient field of $g(x, y)$

Proposition 6.6. *For any (λ_x, λ_y) verifying the two inequalities of Prop. 6.2, $D(\Gamma(\lambda_x, \lambda_y))$ for does not intersect in first quadrant.*

Proof (Proof of Prop. 6.6). For $\Gamma(\lambda_x, \lambda_y) = \Gamma$ satisfying the two inequalities of Prop. 6.2, satisfy: $(1 + \eta)\Gamma > C$ and $C \geq \lambda_x + \lambda_y F > \Gamma$.

Under this, $\frac{da}{d\Gamma}$ and $\frac{db}{d\Gamma}$ are strictly positive and a is strictly positive. Thus, $D(\Gamma(\lambda_x, \lambda_y))$ do not intersect in the first quadrant. □

This basically means g is 'feasible'. As a corollary of Prop. 6.6, we give:

Corollary 6.7. *g can exist in the sense that there are no incompatible constraints resulting from Prop. 6.2.*

Proof (Proof of Corollary 6.7). Prop. 6.5 gives the value g must have on $D(\Gamma)$ in order to have equilibria on it and according to Prop. 6.6 these lines do not intersect in the first quadrant where g can be defined. It proves there is no incompatibility in the definition of g. □

Proposition 6.8

$$\lim_{\Gamma \to C^-} a(\Gamma) = +\infty \; ; \qquad \lim_{\Gamma \to \frac{C}{1+\eta}^+} a(\Gamma) = 0$$
$$\lim_{\Gamma \to C^-} b(\Gamma) = +\infty \; ; \qquad \lim_{\Gamma \to \frac{C}{1+\eta}^+} b(\Gamma) = -1$$

Proof (Proof of Prop. 6.8). Trivial. □

In particular, this last proposition implies that g can be defined in the whole first quadrant using Eq. (13) and lines $D(\Gamma)$.

As of now, we demonstrated that it is feasible to define g so that point of $D(\Gamma)$ can be equilibria for (λ_x, λ_y) on Γ line. Next, we study the stability of the potential equilibria in order to define the additional constraints on m. The only constraint on m coming from existence of equilibrium (Prop. 6.2) is that $m(x_e, y_e) = \frac{C-(\lambda_x+\lambda_y F)}{\eta \lambda_y F}$. The point where this will hold is not specified and depends on m. Defining m will thus define a mapping of arrival rates (λ_x, λ_y) to the actual equilibrium point.

6.2 Stability of the Equilibria and Definition of m

As demonstrated in the previous section, g that doesn't prevent existence of equilibrium is feasible. We now like to have the constraints that m has to satisfy. We already know from the previous section the range m must cover, but we don't know where they have to be located in xy-plane. In order to get more constraints on m, we study the conditions for stable equilibriums. To do so we rely on Hartman Grobman theorem and the study of the stability of the linearized system.

Proposition 6.9. *For an equilibrium point (x_e, y_e) as defined by Prop. 6.2 if the following equations hold:*

$$\begin{cases} \dfrac{\partial m}{\partial x} < \frac{(1+y_e)C}{\lambda_y(F(2+\eta+y_e)+x_e(1+\eta))^2} \\ \dfrac{\partial m}{\partial x} + \left(\dfrac{y_e+1}{x_e+F}\right)\dfrac{\partial m}{\partial y} < 0 \end{cases} \tag{18}$$

then (x_e, y_e) is asymptotically stable.

Proof (Proof of Prop. 6.9)

To analyse of the equilibrium point (x_e, y_e), we take the Jacobian J of the rate equations (2) and (3) at this point. The partial derivatives $\frac{\partial g}{\partial x}$ and $\frac{\partial g}{\partial y}$ at (x_e, y_e) are obtained from the delay equation, Eq. (7).

$$\frac{\partial g}{\partial x}(x_e, y_e) = \frac{(1+\eta)(y_e+1)FC}{(Fy_e + x_e(1+\eta) + F(2+\eta))^2} \tag{19}$$

$$\frac{\partial g}{\partial y}(x_e, y_e) = -\frac{(1+\eta)(x_e+F)FC}{(Fy_e + x_e(1+\eta) + F(2+\eta))^2} \tag{20}$$

The equilibrium point (x_e, y_e) is asymptotically stable if the eigenvalues of the J at (x_e, y_e) have strictly negative real parts [7, Ch. 2 & 5]. Characteristic polynomial of J is:

$$\begin{aligned} &\lambda^2 + \\ &(\lambda_y F(\tfrac{\partial m}{\partial y} - (1+\eta)\tfrac{\partial m}{\partial x}) + \tfrac{\partial g}{\partial x} - \tfrac{\partial g}{\partial y})\ \lambda\ + \\ &\eta(\lambda_y F(\tfrac{\partial m}{\partial x}\tfrac{\partial g}{\partial y} - \tfrac{\partial m}{\partial y}\tfrac{\partial g}{\partial x})) \end{aligned}$$

From this and equations (19) and (20), real parts of the roots are strictly negative iff:

$$(1+\eta)\frac{\partial m}{\partial x} - \frac{\partial m}{\partial y} < \frac{(1+\eta)C(1+x_e+y_e+F)}{\lambda_y(F(2+y_e+\eta)+x_e(1+\eta))^2} \tag{21}$$

and

$$\frac{\partial m}{\partial x} + \left(\frac{y_e+1}{x_e+F}\right)\frac{\partial m}{\partial y} < 0 \tag{22}$$

Inequalities of the proposition are obtained using combination of equations (21) and (22). □

Proposition 6.2 and 6.9 give sufficient conditions on m to define stable equilibria. Next, we prove that there exists m which stabilizes the system for any arrival setting.

Proposition 6.10. *There exists an m satisfying the constraints of propositions 6.2 and 6.9 which stabilizes the system for any arrival rates in the shaded region of the Fig. 4.*

Proof (Proof of Prop. 6.10)

We prove this by exhibiting one such m. Let m be such that

$$m(x, y) = e^{-xy}$$

The m satisfies the constraints of Eq. (18) for any λ_y in $(0, C)$ as $\frac{\partial m}{\partial x} = -ye^{-xy}$ and $\frac{\partial m}{\partial y} = -xe^{-xy}$, are both strictly negative on the interior. Besides, as m ranges from 1 to 0, from the borders ($y = 0$ and $x = 0$) to infinity, thus by continuity, there exists an equilibrium point (x_e, y_e) where $m(x_e, y_e) = \frac{C-(\lambda_x+\lambda_y F)}{\eta\lambda_y F}$ for any arrival rates as all $D(\Gamma)$-lines enter the first quadrant by one its borders. □

Note that if m is strictly monotonic, there is only one equilibrium point for any arrival rate setting in the equilibrium existence region (refer Fig. 3) located at the intersection of the level set of g and m. In addition, it is not possible to apply this for all setting of arrival rates in order to get equilibria for all of them, unless queue are infinite.

Proposition 6.11. *If queues are finite, some setting of arrival rates can't have equilibrium.*

Proof (Proof of Prop. 6.11). As $a(\Gamma)$ tends to 0 when Γ tends to $C/1 + \eta^+$, and $b(\Gamma)$ tends to -1, intersection of $y = 0$ and $D(\Gamma)$ tends to infinity. Hence, for any x_{max}, it is possible to find Γ close enough to $C/1 + \eta$ so that equilibrium which have to be on $D(\Gamma)$ (due to Prop. 6.5) would have to be after x_{max}.

Using limits of $a(\Gamma)$ and $b(\Gamma)$ when Γ tends to C, it is possible to pursue the same reasoning and prove that for some settings of arrival rates, there can't be equilibrium under finite queue for large flows. □

Thus we see that, the system can attain stability depending on the decision of users, and the control function used by operator.

7 Game

We summarize our results in the form of a game with two players: operator and user (with a large flow to send). Here, we make the fair assumption that $T_x < T_y$. The operator can take one of the two actions:

- AFP: Assume fair play, and not use a g.
- AUP: Assume unfair play, and use a g.

From the users, we consider a collective behaviour.

- UC: Users cheat,
- UR: Users rightful

Under AUP, $T_x = T_y$. We use preferential ordering of payoffs for both players. That is $(a_o, a_u) \prec_p (a'_o, a'_u)$, if player p prefers second strategy over the first. The letter o is used to refer to operator, and u to refer to users.

- $(\text{AFP}, \text{UR}) \prec_u (\text{AFP}, \text{UC})$: Users prefer to cheat when the operator does nothing to stop them from cheating, as this would give them shorter response time in the small queue (when $T_x < T_y$).
- $(\text{AUP}, \text{UR}) \prec_u (\text{AUP}, \text{UC})$: Users also prefer to cheat when the operators are aware and are setting service rates dynamically to achieve stability, as this would ensure a finite queue; hence a finite delay. Observe that, if the don't cheat (and stay in Q_y), there is no equilibrium (from Prop. 6.2); hence the queue will build up without bound.

Therefore, it can be drawn that UC strictly dominates UR under any action of the operator (AFP or AUP). Hence, the action UR can be eliminated [8]. So, what lefts to be analysed is the preference of operator under this user action (UR). We see, $(\text{AFP}, \text{UC}) \prec_o (\text{AUP}, \text{UC})$, as there is no equilibrium for general arrival rates (from Prop. 6.2, and if T_x remains less than T_y, migration will create additional load due to η) leading to overflow.

From the above analysis, (AUP, UC) is a Nash equilibrium in the two-players game. That is, assuming operators and users are rational, users will tend to cheat, and operators will look to stabilize the system to maintain finite queues (when the system is operating near to saturation, depending on η).

Note that if the operator's setting of service rates is such that $T_x > T_y$, then migrating to small queue is no more an incentive for large flows. This doesn't preclude operator from favoring small flows as $\frac{x}{\phi_x} < \frac{y}{\phi_y}$ can still hold. In such a scenario, it can be seen that (AFP, UR) will be a Nash equilibrium. This situation can happen if η is large enough and $\frac{\phi_y}{\phi_x}$ can be maintained such that:

$$\begin{cases} \frac{\phi_y}{\phi_x} < \frac{y}{x} \\ \frac{y+1}{x+f}\frac{F}{1+\eta} < \frac{\phi_y}{\phi_x} \end{cases}$$

Second constraint will not be satisfied if operator want to favor small flows too much, say, as in the priority based scheduling proposed in [2]; meaning that it will be of interest for users to cheat.

8 Conclusions

Starting from the setting of static service rates, and moving to dynamic service rate settings, we analysed the existence of equilibria. For the existence of equilibria that is of interest to the operator, it is necessary to have control over the

service rate as a function of the queue lengths. Even then, not all the stable equilibrium points are of interest to the operator, as they give the same delay to small and large flows. Therefore, if a large number of users cheat, the operator has no visible incentive in deploying a size-based scheduling system.

The focus of our study revolved around saturation (of the line capacity) as we assumed that there is some cost η incurred due to migration. In the future, we plan to analyse the system in overload. Similarly, it would be interesting to understand what happens if the operator deploys a mechanism to detect and shift some of the disguised large flows from the small queue to the large queue.

References

1. Kleinrock, L.: Queueing Systems, Volume II: Computer Applications. Wiley Interscience, Hoboken (1976)
2. Avrachenkov, K., Ayesta, U., Brown, P., Nyberg, E.: Differentiation Between Short and Long TCP Flows: Predictability of the Response Time. In: Proc. IEEE INFOCOM (2004)
3. Sun, C., Shi, L., Hu, C., Liu, B.: DRR-SFF: A Practical Scheduling Algorithm to Improve the Performance of Short Flows. In: ICNS 2007: Proceedings of the Third International Conference on Networking and Services, p. 13 (2007)
4. Nuyens, M., Wierman, A.: The foreground-background queue: A survey. Perform. Eval. 65(3-4), 286–307 (2008)
5. Brown, P.: Stability of networks with age-based scheduling. In: Proc. IEEE INFOCOM, pp. 901–909 (2007)
6. Altman, E., Orda, A., Shimkin, N.: Bandwidth allocation for guaranteed versus best effort service categories. Queueing Syst. Theory Appl. 36(1-3), 89–105 (2000)
7. Sastry, S.: Nonlinear Systems. Analysis, Stability and Control. Springer, Heidelberg (1999)
8. Osborne, M.J., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge (1994)

Scalable Model for Packet Loss Analysis of Load-Balancing Switches with Identical Input Processes

Yury Audzevich[1], Levente Bodrog[2], Yoram Ofek[1], and Miklós Telek[2]

[1] Department of Information Engineering and Computer Science,
University of Trento, Italy
{audzevi,ofek}@disi.unitn.it
[2] Department of Telecommunications,
Technical University of Budapest, Hungary
{bodrog,telek}@hit.bme.hu

Abstract. In this paper we present a scalable approximate model for packet loss analysis in load-balancing Birkhof-von Neumann switch with finite buffers and variable length packets assumption. We also present a numerical method to solve the model for large switches (up to the size ~ 30) equipped with large buffers (up to the buffer size ~ 1000). With regards to previously introduced models the main contribution of our model is its scalability in terms of the switch size as its computational complexity is linear with the number of ports. Contrary to previous models we assumed homogeneous input processes in this paper.

1 Introduction

Internet is a huge asynchronous mesh network which is composed of several sub-networks connected to each other through switches. As the traffic over the network and the number of links grow exponentially the transmitting media can be easily adopted using optical fibre. Although the links could provide high throughput, the switches are not always capable to fulfill both the throughput growth and increasing number of connections. Some solutions with high throughput and centralized control exist but they are poorly scalable.

Recently in [1,2] the authors introduced a promising and highly scalable solution, a two-stage switching architecture called load-balancing (LB) Birkhof-von Neumann switch.

[1,2] shows initial investigations on the switch under some strong assumptions (infinite buffers, traffic admissibility, equal size packets in the system). On the contrary [3] used realistic scenarios and carried out a simulation based throughput analysis of the LB switch with finite buffers. [4] pointed out that in cell-based (packets of the same size) LB switch a loss can occur because of buffer overflow. This latter paper also presents mathematical analysis for cell loss probability evaluation. Besides, going further in this approach, [5,6] give analytical results for both finite buffers and variable size packets.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 249–263, 2009.

Whereas in [5] the authors present the full characterization of a realistic scenario, with finite buffers and variable size packets, a less complex approximating model is given in [6]. In spite of the complexity $O\left(2^N\right)$ we still need a fast procedure to solve the model. The aim of this paper is to present an analysis with fast solution procedure. However a restrictive assumption is applied, i.e., the model assumes identical stochastic processes on all the inputs.

We will demonstrate that, besides this assumption, the newly introduced model captures the two most important performance measures. We analyzed the packet loss – as the switch is equipped with finite buffers – and gave an estimate of the mean packet waiting time. The first parameter affects the Quality of Service (QoS) characteristics of data transfers (using TCP). The second parameter has high influence on real time traffic, e.g., speech (using UDP) over the network [7,8].

We also introduced a folding algorithm-based numerical method to solve the model of switches with large buffers.

The rest of the paper is organized as follows. In Section 2 we give the modeling assumptions and the basic principles of the switch. Section 3 presents the model into detail. The numerical solution method is introduced in Section 4 and Section 5 verifies the model. Finally Section 6 concludes the paper.

2 Basic Principles and Main Assumptions of the Switching Mechanism

2.1 Basic Principles

The LB switch is considered to be a two-stage switching architecture. The first stage uniformly distributes the arriving traffic to the central stage, which is an input buffer of the second switch (see Figure 1). Its scalability lies in the distributed, distinct and deterministic control between different switching stages.

To improve the buffer utilization the arriving packets are segmented into cells of equal size. The basic operating time unit is the service time of a cell – hereinafter referred to as *time slot.*

The arrival rate of the cells at the input ports are assumed to be identical to the service rate of the inputs, i.e., there is no cell loss here. The service rate of

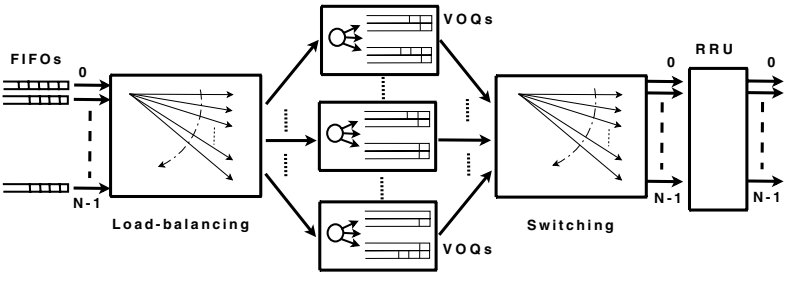


Fig. 1. The load-balancing switch considered for the analysis

each output is assumed to be greater than the arrival rate of cell, i.e., the switch is not overloaded.

In the followings N denotes the size of the switch, i.e., the number of input and output ports. The central stage consists of N sets of N virtual output queues (VOQs). In each set there is one buffer dedicated to every output. Hereinafter VOQ_k denotes the kth set of VOQs. The cells directed to output j are put in the jth VOQ out of the kth set – hereinafter denoted as VOQ_{kj}.

During the t_1st time slot input i is connected to VOQ_k according to round-robin (RR) interconnection policy

$$k = i + t_1 \mod N \quad i, k \in [0, N-1] \tag{1}$$

by means of crossbar switches without buffers inside (contrary to [9]). The actual cell arriving from input i and directed to output j is put into VOQ_{kj} if a free position is available and it is dropped otherwise. In our assumption a cell can only be lost due to buffer overflow as the VOQs are finite. The VOQs are served according to the FIFO policy.

As the packets are segmented into cells we consider a packet to be lost, when at least one of its cells is lost, i.e., packet loss can occur also according to the finite VOQs.

VOQ_{kj} is served in the t_2nd time slot, when VOQ_k is connected to output j by means of crossbar switches operating according to RR policy

$$j = k + t_2 \mod N \quad j, k \in [0, N-1]\,. \tag{2}$$

As both crossbars applies RR interconnection policy with the same modulus (N), the LB switch itself has periodic behavior of period N time slots – hereinafter referred to as *time period.*

Finally a packet is reassembled in the re-sequencing and reassembly unit (RRU) at the output (see Figure 1), like in [10], and sent to the external link.

2.2 Modeling Assumptions

In a time slot, first, the VOQs are connected to the outputs and then the inputs to the VOQs. This order of interconnections inhibits a cell from passing through the switch in a single time slot.

During our work we assumed Markovian behavior of system, more precisely geometric distributed random variables. On the one hand we can fit one parameter of the observed distributions, but on the other hand we can use the sophisticated and numerically efficient algorithms to solve discrete time Markov chains (DTMCs). In order to increase the precision of the analysis, one can expand the number of fitted parameters to an arbitrary level by using more complex Markovian structures like discrete time Phase-Type (DPH) distributions or discrete time Markovian Arrival Processes (DMAPs). Yet such a choice would increase the complexity of the model and, to a certain extent, shift the focus of the paper.

According to the Markovian assumption the packet length (X) distribution (in cells) of the arrival process is geometric distributed with probability mass function (PMF)

$$\Pr(X = i) = \hat{p}\,(1 - \hat{p})^{i-1} \quad i = 1, 2, \ldots \tag{3}$$

The length of the idle periods between packets (Y) are also geometric distributed (in time slots) with PMF

$$\Pr(Y = i) = \hat{q}\,(1 - \hat{q})^{i} \quad i = 0, 1, \ldots \tag{4}$$

The parameters are the same for all inputs according to the identical input process assumption, which makes us possible to introduce a compact approximate model of the LB switch. The packets arriving at an arbitrary input are spread uniformly between the outputs, i.e., the probability of sending a packet to a particular output is

$$\hat{t} = \frac{1}{N}. \tag{5}$$

Previous works [5,6] introduced the differences between traffic paths traversing the switch. This phenomenon is recalled in the next section.

2.3 On the Different Paths

It is shown in [5] and [6] that the cell loss probability and accordingly the packet loss probability depend on the path through which it traverses the switch. Where path means a triple, denoted as $\{i, j, k\}$, containing the ordinal number of the input, the output and the VOQ respectively.

Mainly the difference of the paths comes from the time difference between the service of a VOQ and the arrival to it. Using (1) and (2) the time difference between the service of a VOQ and the arrival to it is expressed as

$$t_2 - t_1 = d = 2k - i - j \mod N, \tag{6}$$

which also gives the number of inputs that have the right to send cells to VOQ_{kj} before input i in the same time period. d is then directly proportional to the loss probability of a path, i.e., the higher the d value is the higher the loss probability of that path is. Here we use the term *loss probability of a path* to emphasize the difference between the cell loss probabilities depending on the triple $\{i, j, k\}$ or equivalently depending on d.

Based on (6) we recall the term type-d path introduced in [6] for a given path with characteristic value d.

2.4 On the Definition of the Different Loss Probabilities

Cell loss probability is simply the ratio between the number of cells which were dropped from the observed VOQ versus the total number of cells sent through the VOQ.

The calculation of a *packet loss* inside the VOQ is not that trivial since cells belonging to the same packet can be spread to different VOQs. Accordingly we consider the packet loss in terms of a specific VOQ, i.e., the packet is considered to be lost, if at least one of its cells is lost in the observed VOQ.

The cell loss probability and accordingly the packet loss probability depends on the path as it is described in the previous section, it is referred as *loss probability of a path.*

Since our main interest is the packet loss probability, the precise way how it is calculated is given in Section 3.4 and the different loss probabilities for all the paths are considered in the numerical study in Section 5.

3 The Model

In this section we give the detailed model of VOQ_{00} as part of path $\{1,0,0\}$ of the 3×3 switch. This is a type-2 path of that particular switch, but the detailed analysis of all 3 types of paths will be given in Section 3.4.

3.1 The Model of the Input Processes

The parameters of the identical input process are

$\hat{p}$ the parameter of the geometric distributed packet length (3) in cells,
$\hat{q}$ the parameter of the geometric distributed idle period length (4) in time slots and
$\hat{t}=\frac{1}{N}$ the probability of choosing a specific output for a given packet (5).

Based on the geometric assumption we can build the DTMC model, fully characterizing any of the identical inputs, with state transition probability matrix

$$\mathbf{P}^{\mathcal{C}}=\begin{pmatrix}(1-\hat{p})+\hat{p}\hat{q}\hat{t} & \hat{p}\hat{q}\hat{t} & \hat{p}\hat{q}\hat{t} & \hat{p}\,(1-\hat{q})\\ \hat{p}\hat{q}\hat{t} & (1-\hat{p})+\hat{p}\hat{q}\hat{t} & \hat{p}\hat{q}\hat{t} & \hat{p}\,(1-\hat{q})\\ \hat{p}\hat{q}\hat{t} & \hat{p}\hat{q}\hat{t} & (1-\hat{p})+\hat{p}\hat{q}\hat{t} & \hat{p}\,(1-\hat{q})\\ \hat{q}\hat{t} & \hat{q}\hat{t} & \hat{q}\hat{t} & 1-\hat{q}\end{pmatrix} \tag{7}$$

and graph given in Figure 2, where the state identifiers are the following

j corresponds to cell arrival from the input to output j $\quad j=0,1,2$
id corresponds to the idle period of the input.

According to the observed output, i.e., output 0, the states of the DTMC in Figure 2 are divided into two subsets, a one element subset and all the others, hereinafter denoted as **on** and **off**, respectively. Their meaning are

on the state represents cell arrival from the observed input to output 0 and
off the states represent no cell arrival from the observed input to output 0.

In the following we introduce the approximating two state ON/OFF model of the general input mainly as replacing the set **off** with a single state OFF. Hereinafter uppercase ON and OFF denote the states of the approximating two state description of the input process.

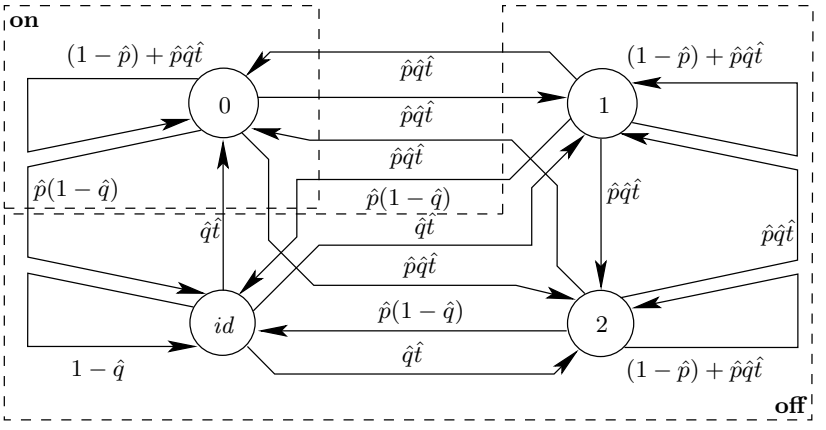


Fig. 2. The graph of the DTMC fully characterizing any input of the 3×3 switch

The ON Properties. State ON replaces the one element subset **on** with the same sojourn probability $(1-\hat{p})+\hat{p}\hat{q}\hat{t}$. Accordingly the state transition probability from ON to OFF is 1 minus the sojourn probability $\hat{p}-\hat{p}\hat{q}\hat{t}$.

The OFF Properties. The OFF state replaces the set of **off** states by approximating their sojourn time with the absorbing time of a DPH distribution described in the followings.

For output 0 the transient states of the DPH are the **off** states and the absorbing state is the **on** state as depicted in Figure 3.

Based on $\mathbf{P}^{\mathcal{C}}$, given in (7), we give the initial distribution ($\boldsymbol{\beta}$) and the state transition probability matrix ($\mathbf{B}$) of the DPH. The initial distribution is the state probability right after entering **off** from **on**. It is obtained as the renormalization of the zeroth row of $\mathbf{P}^{\mathcal{C}}$ without its zeroth element

$$\boldsymbol{\beta} = \left(\frac{\hat{q}\hat{t}}{2\hat{q}\hat{t}+(1-\hat{q})} \;\; \frac{\hat{q}\hat{t}}{2\hat{q}\hat{t}+(1-\hat{q})} \;\; \frac{1-\hat{q}}{2\hat{q}\hat{t}+(1-\hat{q})} \right),$$

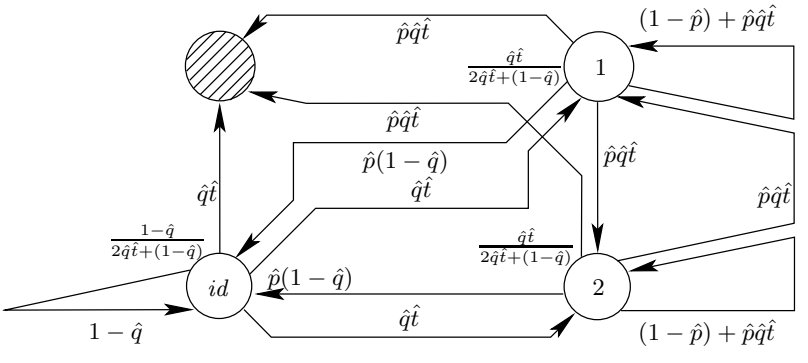


Fig. 3. The graph of the DPH substitution of the **off** states in terms of output 0

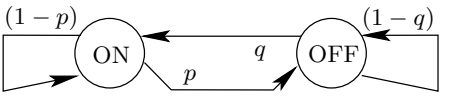


Fig. 4. The ON/OFF model of the input process with the simplified notation

which is also indicated in Figure 3. The 3×3 sized state transition probability matrix of the **off** states is obtained from $\mathbf{P}^{\mathcal{C}}$ by cutting its zeroth row and zeroth column

$$\mathbf{B} = \begin{pmatrix} (1-\hat{p}) + \hat{p}\hat{q}\hat{t} & \hat{p}\hat{q}\hat{t} & \hat{p}\,(1-\hat{q}) \\ \hat{p}\hat{q}\hat{t} & (1-\hat{p}) + \hat{p}\hat{q}\hat{t} & \hat{p}\,(1-\hat{q}) \\ \hat{q}\hat{t} & \hat{q}\hat{t} & 1-\hat{q} \end{pmatrix}.$$

The mean absorbing time of this DPH is then

$$\mu = \boldsymbol{\beta}\,(\mathbf{I} - \mathbf{B})^{-1}\,\mathbf{h}, \tag{8}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{h}$ is the column vector of ones of appropriate size.

Here we note that according to the structure of (7) μ is the same for any output and any input – indeed the input processes are identical.

Consequently the sojourn probability of state OFF is $1 - \frac{1}{\mu}$. The state transition probability from OFF to ON is $\frac{1}{\mu}$ which sets the mean sojourn time in state OFF equal to μ.

The state transition probability matrix of the two state DTMC describing the ON/OFF input process for the general path is

$$\mathbf{P} = \begin{pmatrix} (1-\hat{p}) + \hat{p}\hat{q}\hat{t} & \hat{p} - \hat{p}\hat{q}\hat{t} \\ \frac{1}{\mu} & 1 - \frac{1}{\mu} \end{pmatrix} = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}, \tag{9}$$

where we also introduced a simplified notation with p and q. The graph of the ON/OFF DTMC using the simplified notation is given in Figure 4 which is the same for all the inputs according to the identical input process assumption.

3.2 Aggregate Input Model

We describe the combined behavior of the N inputs by a DTMC of $N+1$ states representing the number of inputs in ON. Using the considerations in Section 3.1 and especially (9) the ijth element of the state transition probability matrix of such a DTMC describing N inputs after 1 time slots is

$$\left(\boldsymbol{\mathcal{P}}_{N,1}(p,q)\right)_{ij} = \sum_{k=\max(0,j-i)}^{\min(i,N-j)} \binom{i}{k} p^k\,(1-p)^{i-k} \binom{N-i}{j-i+k} q^{j-i+k}\,(1-q)^{N-j-k} \tag{10}$$

where we also indicated that these probabilities depend on the parameters of (9) – p, q. The first binomial factor of (10) represents that out of i ON sources k

moves to OFF and the second factor represents that out of $N-i$ OFF sources $j-i+k$ moves to ON, $i,j \in [0, N-1]$. (10) also introduces the notation $\boldsymbol{\mathcal{P}}_{N,M}(p,q)$ hereinafter denoting the state of N inputs during M time slots with each input modeled by an ON/OFF DTMC with parameters p and q given in (9). For example the state of N inputs after M time slots is

$$\boldsymbol{\mathcal{P}}_{N,M}(p,q) = \boldsymbol{\mathcal{P}}^{M}_{N,1}(p,q). \tag{11}$$

Using the above method there can be given behavior of any number of inputs in any number of time slots.

Based on $\boldsymbol{\mathcal{P}}_{N,M}(p,q)$ we give the arrival based decomposition of the arrival process as

$$\underbrace{\mathbf{B} = \begin{pmatrix} \mathbf{p}^0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{\text{0 arrivals}} \quad \underbrace{\mathbf{L} = \begin{pmatrix} 0 \\ \mathbf{p}^1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}}_{\text{1 arrival}} \quad \underbrace{\mathbf{F}_1 = \begin{pmatrix} 0 \\ 0 \\ \mathbf{p}^2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{\text{2 arrivals}} \quad \cdots \quad \underbrace{\mathbf{F}_{N-1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{p}^N \end{pmatrix}}_{\text{N arrivals}}, \tag{12}$$

where $\mathbf{p}^i$ denotes the ith row vector of $\boldsymbol{\mathcal{P}}_{N,M}(p,q)$.

The arrival based decomposition of the $N \times N$ switch in M time slots, is formalized in Algorithm 1.

Algorithm 1. Arrival based decomposition of the input process

INPUT: $N, M, \mathbf{P}$ from (9)
OUTPUT: $\mathbf{B}, \mathbf{L}, \mathbf{F}_1, \ldots, \mathbf{F}_{N-1}$ the arrival based decomposition
1. determine $\boldsymbol{\mathcal{P}}_{N,M}(p,q)$ similar to (11) using $\mathbf{P}$
2. decompose $\boldsymbol{\mathcal{P}}_{N,M}(p,q)$ as in (12)
3. **return** $\mathbf{B}, \mathbf{L}, \mathbf{F}_1, \ldots, \mathbf{F}_{N-1}$

3.3 The Cell Level Model of the 3 × 3 Switch

We show how the packet loss is calculated in path $\{1,0,0\}$ for which we give the cell level model of the corresponding VOQ – VOQ_{00}. It is a quasi birth-death like (QBD-like) structure whose level represents the queue length and phase represents the state of the input process.

As the phase process of the QBD-like model is the combined state of the inputs their arrival based decomposition gives the level transition matrices used to build the QBD-like structure. $\mathbf{B}, \mathbf{L}, \mathbf{F}_1, \mathbf{F}_2$ are determined by Algorithm 1 with input parameters $N = 3$, according to the number of inputs, $M = 3$ the number of time slots in a time period and $\mathbf{P}$ (from (9)). Here $M = 3$ since the time period of the DTMC is 3 time slots long – as it is given in Section 2.1.

A level transition backward is according to $\mathbf{B}$ since there is one cell served during a time period and $\mathbf{B}$ represents 0 arrivals. Local state transition is according to $\mathbf{L}$ and there are 1(2) forward level transition(s) according to $\mathbf{F}_1(\mathbf{F}_2)$.

The state transition probability matrix of the QBD-like model is

$$\mathbb{P} = \begin{pmatrix} \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & 0 & \dots \\ \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 \\ \dots & 0 & 0 & \mathbf{B} & \mathbf{L} & \mathbf{F}'_1 \\ \dots & 0 & 0 & 0 & \mathbf{B} & \mathbf{L}' \end{pmatrix}, \tag{13}$$

where $\mathbf{F}'_1 = \mathbf{F}_1 + \mathbf{F}_2$ and $\mathbf{L}' = \mathbf{L} + \mathbf{F}_1 + \mathbf{F}_2$.

The steady state solution of this QBD-like model is the solution of the linear system of equations

$$\boldsymbol{\pi}\mathbb{P} = \boldsymbol{\pi}, \qquad \boldsymbol{\pi}\mathbf{h} = 1. \tag{14}$$

3.4 Packet Level Model

With the geometric assumption for the packet length, given in Section 2.2, the life cycle of a packet in the observed path can be modeled by a transient DTMC in which there are two absorbing states corresponding to the two possible ending of a packet. The first absorbing state corresponds to the first cell loss, or equivalently the packet loss (PL) and the other one corresponds to the successful packet transmission (ST). The transient DTMC with two absorbing states is given in Figure 5. In this section we present this transient DTMC with its state transition probability matrix and initial distribution based representation.

The Transient Part of the DTMC. Basically during the life cycle of a packet VOQ_{00} is modeled by a quasi birth like (QB-like) structure. Its level represents

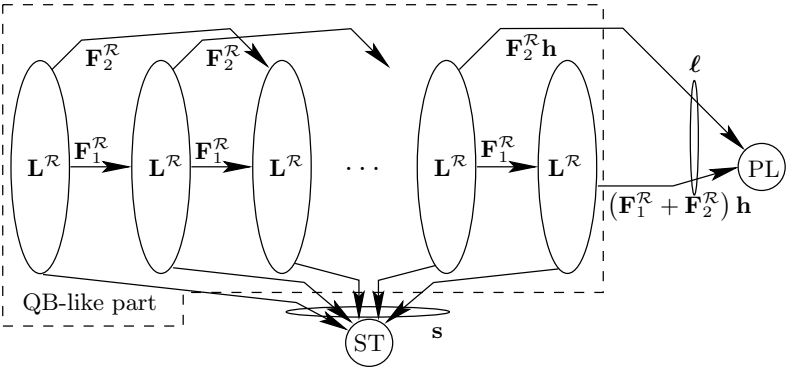


Fig. 5. The transient DTMC modelling the VOQ during the life cycle of a packet

the queue length and its phase process is the combined state of the 3 inputs. In this case there is one important difference compared to the model given in the previous section. Input 1 is in ON for sure, since this is the model of the life cycle of a packet arrives from input 1, which also implies that there is no backward level transition.

The other two inputs behave in the "normal" manner, i.e., their corresponding level transition matrices are determined by Algorithm 1 with input parameters $N = 2$, $M = 3$ and $\mathbf{P}$ in (9). $M = 3$ since the time unit of the 3×3 switch is 3 time slots. The result of the algorithm is

$$\mathbf{B}, \mathbf{L} \text{ and } \mathbf{F}, \tag{15}$$

of size 3×3 as they describe 2 inputs (the possible states of this phase process are $0, 1$ and 2 – the number of inputs that are in ON).

According to these considerations the state transition probability matrix of the QB-like structure is built using the blocks

$$\mathbf{L}^{\mathcal{R}} = (1-p)^3 \mathbf{B}, \qquad \mathbf{F}_1^{\mathcal{R}} = (1-p)^3 \mathbf{L} \qquad \text{and} \qquad \mathbf{F}_2^{\mathcal{R}} = (1-p)^3 \mathbf{F}. \tag{16}$$

Superscript $\mathcal{R}$ denotes quantities describing this transient DTMC of Figure 5. (16) describes the joint behavior of input 1 (given by $(1-p)^3$, the probability that input 1 remains in ON) and the other two inputs (given by matrices $\mathbf{B}, \mathbf{L}, \mathbf{F}$).

Finally using (16) the state transition probability matrix of the transient part and the state transition probability vector to state PL are

$$\mathbb{P}^{\mathcal{R}} = \begin{pmatrix} \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} \\ \dots & 0 & 0 & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} \\ \dots & 0 & 0 & 0 & \mathbf{L}^{\mathcal{R}} \end{pmatrix}, \qquad \boldsymbol{\ell} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{F}_2^{\mathcal{R}} \mathbf{h} \\ \left(\mathbf{F}_1^{\mathcal{R}} + \mathbf{F}_2^{\mathcal{R}}\right) \mathbf{h} \end{pmatrix}, \tag{17}$$

where $\boldsymbol{\ell}$ tells that if input 1 is in ON (which is the fundamental assumption here) then there is packet loss if at the beginning of the time period there is either

- one free position in the VOQ and there are three arrivals $\left(\mathbf{F}_2^{\mathcal{R}} \mathbf{h}\right)$ or
- no free positions in the buffer and there are either
  - two arrivals $\left(\mathbf{F}_1^{\mathcal{R}} \mathbf{h}\right)$ or
  - three arrivals $\left(\mathbf{F}_2^{\mathcal{R}} \mathbf{h}\right)$.

Using $\mathbb{P}^{\mathcal{R}} \mathbf{h} + \boldsymbol{\ell} + \mathbf{s} = \mathbf{h}$ the state transition probability vector to state ST is

$$\mathbf{s} = \mathbf{h} - \left(\mathbb{P}^{\mathcal{R}} \mathbf{h} + \boldsymbol{\ell}\right). \tag{18}$$

The Initial Distribution of the Transient DTMC. The initial distribution of $\mathbb{P}^{\mathcal{R}}$ in (17) is determined as the state of the system right after the arrival of an incoming packet. In this section we determine the probability distribution of the system at this time instance, right after a new packet arrival.

Here we give the joint probability of arriving a new packet at input 1 and the "normal" behavior of the other two inputs. Using the notations introduced in (9) the first probability is $1-(1-q)^3$ and latter one is determined as the output of Algorithm 1 with input parameters $N=2, M=3, \mathbf{P}$, the same as in (15). If $\tilde{q}=1-q$ then their joint behavior is described by the matrices

$$\hat{\mathbf{B}}^{\mathcal{N}}=\left(1-\tilde{q}^3\right)\mathbf{B}, \qquad \hat{\mathbf{L}}^{\mathcal{N}}=\left(1-\tilde{q}^3\right)\mathbf{L} \qquad \text{and} \qquad \hat{\mathbf{F}}^{\mathcal{N}}=\left(1-\tilde{q}^3\right)\mathbf{F}. \tag{19}$$

The block sizes of $\boldsymbol{\pi}$ in (14) are 4 since they describe all the 3 inputs. According to this there is a row of zeros appended to every level transition matrices in (19) as

$$\mathbf{B}^{\mathcal{N}}=\begin{pmatrix}\hat{\mathbf{B}}^{\mathcal{N}}\\ 0\end{pmatrix} \qquad \mathbf{L}^{\mathcal{N}}=\begin{pmatrix}\hat{\mathbf{L}}^{\mathcal{N}}\\ 0\end{pmatrix} \qquad \mathbf{F}^{\mathcal{N}}=\begin{pmatrix}\hat{\mathbf{F}}^{\mathcal{N}}\\ 0\end{pmatrix}. \tag{20}$$

The last row expresses that in case of a new packet arrival there cannot be all the $N=3$ inputs in ON. Here we recall that in our model there is no corresponding cell arrival to state change from OFF to ON, i.e., in case of new packet arrival there is no cell arrival from the observed input.

Then starting from the steady state of the cell level model (14) and using the level transitions according to new packet arrival (20) the blocks of the initial distribution of the transient DTMC given in Figure 5 are

$$\begin{aligned}
\hat{\boldsymbol{\pi}}_0^{\mathcal{N}} &= \boldsymbol{\pi}_0\mathbf{B}^{\mathcal{N}}+\boldsymbol{\pi}_1\mathbf{B}^{\mathcal{N}}\\
\hat{\boldsymbol{\pi}}_1^{\mathcal{N}} &= \boldsymbol{\pi}_0\mathbf{L}^{\mathcal{N}}+\boldsymbol{\pi}_1\mathbf{L}^{\mathcal{N}}+\boldsymbol{\pi}_2\mathbf{B}^{\mathcal{N}}\\
\hat{\boldsymbol{\pi}}_2^{\mathcal{N}} &= \boldsymbol{\pi}_0\mathbf{F}^{\mathcal{N}}+\boldsymbol{\pi}_1\mathbf{F}^{\mathcal{N}}+\boldsymbol{\pi}_2\mathbf{L}^{\mathcal{N}}+\boldsymbol{\pi}_3\mathbf{B}^{\mathcal{N}}\\
\hat{\boldsymbol{\pi}}_i^{\mathcal{N}} &= \boldsymbol{\pi}_{i-1}\mathbf{F}^{\mathcal{N}}+\boldsymbol{\pi}_i\mathbf{L}^{\mathcal{N}}+\boldsymbol{\pi}_{i+1}\mathbf{B}^{\mathcal{N}} \quad 3\le i\le b-1\\
\hat{\boldsymbol{\pi}}_b^{\mathcal{N}} &= \boldsymbol{\pi}_{b-1}\mathbf{F}^{\mathcal{N}}+\boldsymbol{\pi}_b\left(\mathbf{L}^{\mathcal{N}}+\mathbf{F}^{\mathcal{N}}\right).
\end{aligned}$$

$\hat{\boldsymbol{\pi}}^{\mathcal{N}}$ is normalized as

$$\boldsymbol{\pi}^{\mathcal{N}}=\frac{\hat{\boldsymbol{\pi}}^{\mathcal{N}}}{\hat{\boldsymbol{\pi}}^{\mathcal{N}}\mathbf{h}} \tag{21}$$

resulting in the initial distribution of the packet level model in Figure 5.

The Packet Loss of the System. Using (17), (18) and (21) the packet loss probability of the system and the probability of successful packet transmission on the given path are calculated as absorbing in state PL and ST, respectively, i.e.,

$$p_\ell=\boldsymbol{\pi}^{\mathcal{N}}\left(\mathbf{I}-\mathbb{P}^{\mathcal{R}}\right)^{-1}\boldsymbol{\ell}, \qquad p_s=\boldsymbol{\pi}^{\mathcal{N}}\left(\mathbf{I}-\mathbb{P}^{\mathcal{R}}\right)^{-1}\mathbf{s}=1-p_\ell. \tag{22}$$

Estimation for the Packet Waiting Time. We estimate the mean packet waiting time with the mean cell waiting time. The mean cell waiting time equals to the mean system time of the cells entering the queue minus the cell service time. Since the service of the VOQ is deterministic the system time of a cell in

the VOQ is $N = 3$ time slots times the queue length right after the cell arrival given that the cell is not dropped (denoted as $\tilde{\boldsymbol{\pi}}'$).

$\tilde{\boldsymbol{\pi}}'$ can be determined by the equation system

$$\begin{aligned}
\tilde{\boldsymbol{\pi}}'_1 &= \boldsymbol{\pi}_1 \left(\frac{1}{3}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1 + \mathbf{L}\right) + \boldsymbol{\pi}_0 \left(\frac{2}{3}\mathbf{F}_2 + \mathbf{F}_1 + \mathbf{L}\right) \\
\tilde{\boldsymbol{\pi}}'_i &= \boldsymbol{\pi}_{i-2}\frac{1}{3}\mathbf{F}_2 + \boldsymbol{\pi}_{i-1}\left(\frac{1}{3}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1\right) + \boldsymbol{\pi}_i \left(\frac{1}{3}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1 + \mathbf{L}\right) \quad i \in [2, b-2] \\
\tilde{\boldsymbol{\pi}}'_{b-1} &= \boldsymbol{\pi}_{b-3}\frac{1}{3}\mathbf{F}_2 + \boldsymbol{\pi}_{b-2}\left(\frac{1}{3}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1\right) + \boldsymbol{\pi}_{b-1}\left(\frac{1}{2}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1 + \mathbf{L}\right) \\
\tilde{\boldsymbol{\pi}}'_b &= \boldsymbol{\pi}_{b-2}\frac{1}{3}\mathbf{F}_2 + \boldsymbol{\pi}_{b-1}\left(\frac{1}{2}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1\right) + \boldsymbol{\pi}_b \left(\mathbf{F}_2 + \mathbf{F}_1 + \mathbf{L}\right).
\end{aligned}$$

$\tilde{\boldsymbol{\pi}}'$ is normalized as $\tilde{\boldsymbol{\pi}} = \frac{\tilde{\boldsymbol{\pi}}'}{\tilde{\boldsymbol{\pi}}'\mathbf{h}}$ resulting in the queue length distribution right after a cell arrival given that the cell enters the queue.

4 On the Solution of Large QBD-Like DTMCs

Building the cell level DTMC as in (13) and solving it as in (14) results in the solution of a linear equation system of size $(b+1)(N+1)$ which can be easily led to inaccurate numerical results.

As a fast and numerically efficient solution of this we apply the folding algorithm, e.g., in [11], based solution of (14). The algorithm is prepared to block tri-diagonal matrices, hence we repartition (13) as

$$\mathbb{P} = \begin{pmatrix}
\begin{array}{|cc|cc|}\hline \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 \\ \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 \\ \hline \end{array} & & 0 & \\
\begin{array}{|cc|}\hline 0 & \mathbf{B} \\ 0 & 0 \\ \hline \end{array} & \begin{array}{|cc|}\hline \mathbf{L} & \mathbf{F}_1 \\ \mathbf{B} & \mathbf{L} \\ \hline \end{array} & \begin{array}{|cc|}\hline \mathbf{F}_2 & 0 \\ \mathbf{F}_1 & \mathbf{F}_2 \\ \hline \end{array} & 0 \\
& \ddots & \ddots & \ddots \\
0 & & \begin{array}{|cc|}\hline 0 & \mathbf{B} \\ 0 & 0 \\ \hline \end{array} & \begin{array}{|cc|}\hline \mathbf{L} & \mathbf{F}'_1 \\ \mathbf{B} & \mathbf{L}' \\ \hline \end{array}
\end{pmatrix} = \begin{pmatrix}
\mathbb{L}' & \mathbb{F}' & 0 \\
\mathbb{B} & \mathbb{L} & \mathbb{F} & 0 \\
& \ddots & \ddots & \ddots \\
0 & & \mathbb{B} & \mathbb{L}''
\end{pmatrix}, \tag{23}$$

where we have enlarged the block size to $(N-1)(N+1)$. The inverse of the enlarged block is calculated in the folding algorithm, by which we have increased the complexity as well. We can increase the buffer size b to high values as the computational complexity of the folding algorithm is $O(\log_2 b)$.

In the followings we give the reduction of the matrix inversion of $\mathbf{I} - \mathbb{P}^{\mathcal{R}}$, in (22), to the inversion of its diagonal block – denoted as $\mathbf{V} = \mathbf{I} - \mathbf{L}^{\mathcal{R}}$. Considering the matrix equation

$$\mathbf{x}\left(\mathbf{I} - \mathbb{P}^{\mathcal{R}}\right) = \boldsymbol{\pi}^{\mathcal{N}} \tag{24}$$

where the coefficient matrix $\left(\mathbf{I} - \mathbb{P}^{\mathcal{R}}\right)$ has an upper triangular structure, on the block level, we can apply the following iterative solution of the matrix equation

$$\mathbf{x}_0\mathbf{V} = \boldsymbol{\pi}_0^{\mathcal{N}} \quad \rightarrow \mathbf{x}_0 = \boldsymbol{\pi}_0^{\mathcal{N}}\mathbf{V}^{-1}$$
$$\mathbf{x}_0\mathbf{F}_1 + \mathbf{x}_1\mathbf{V} = \boldsymbol{\pi}_1^{\mathcal{N}} \quad \rightarrow \mathbf{x}_1 = \left(\boldsymbol{\pi}_1^{\mathcal{N}} - \mathbf{x}_0\mathbf{F}_1\right)\mathbf{V}^{-1}$$

and all the other blocks for $i = 2, \ldots, b$ are

$$\mathbf{x}_{i-2}\mathbf{F}_2 + \mathbf{x}_{i-1}\mathbf{F}_1 + \mathbf{x}_i\mathbf{V} = \boldsymbol{\pi}_i^{\mathcal{N}} \quad \rightarrow \mathbf{x}_i = \left(\boldsymbol{\pi}_i^{\mathcal{N}} - \mathbf{x}_{i-1}\mathbf{F}_1 - \mathbf{x}_{i-2}\mathbf{F}_2\right)\mathbf{V}^{-1}$$

Rearranging (24) results in $\mathbf{x} = \boldsymbol{\pi}^{\mathcal{N}}\left(\mathbf{I} - \mathbb{P}^{\mathcal{R}}\right)^{-1}$ which implies that from (22) the packet loss probability (p_ℓ) and the probability of successful packet transmission (p_s) of the observed VOQ can be calculated as

$$p_\ell = \mathbf{x}\boldsymbol{\ell} \qquad \text{and} \qquad p_s = \mathbf{x}\mathbf{s}. \tag{25}$$

5 A Numerical Study

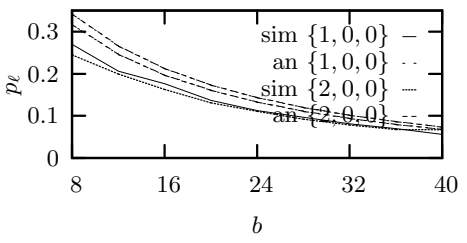
In contrast to [5,6] where we described extended methodology of packet loss analysis in the LB switch, this paper presents optimized solution with linear complexity. The computational study has two parts. The first part shows the behavior of the packet loss and waiting time of the LB switch as a function of buffer length and switch size. The second part examines some extreme cases when central stage buffers are large to show the power of the folding algorithm based solution method presented in Section 4. For the results of this section we used the parameters given in Table 1. In order the comparative analysis, we made the specified measurements also with our LB switch simulation tool.

Table 1. Parameters used for the numerical studies

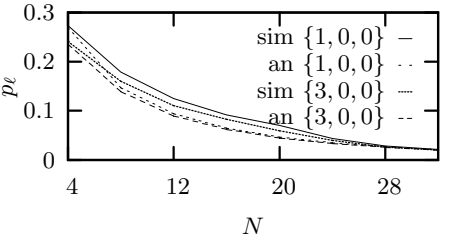
Figure	6(a)	6(b)	6(c)	6(d)	7(a)	7(b)
name	p_ℓ vs. b	p_ℓ vs. N	T vs. b	T vs. N	p_ℓ vs. b	T vs. b
	without folding algorithm				with folding	
N	4	$4, \ldots, 32$	4	$3, \ldots, 33$	3	
b	$8, \ldots, 40$	36	$8, \ldots, 40$	127	$9, \ldots, 999$	
$\hat{p}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{20}$	$\frac{1}{50}$		
$\hat{q}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{3}$			
$\hat{t}$	$\frac{1}{N}$					

Part 1. In [5,6] we examined the dependence of packet loss at the central stage buffers on the buffer size and switch size. It was found that the packet loss probability strongly depends on the chosen path ($\{i, j, k\}$). Figure 6(a) and 6(b) present similar results using the approximate model introduced in this paper.

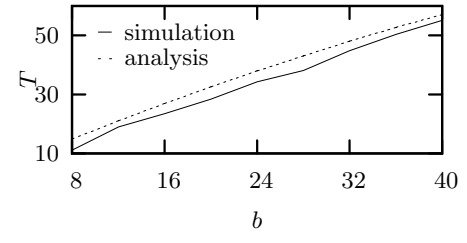
Figures 6(c) and 6(d) indicate another performance characteristic, the packet waiting time estimator compared to simulation results. The packet waiting time is evaluated considering only the successfully transmitted packets. The packet waiting time is generally increases together with the buffer size (larger interval between cell arrivals and services), like in Figure 6(c) and switch size (cells are spread to more queues), like in Figure 6(d).



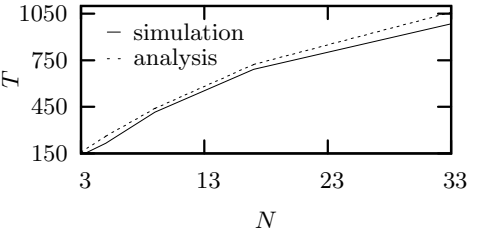
(a) Packet loss versus the buffer size



(b) Packet loss versus the switch size

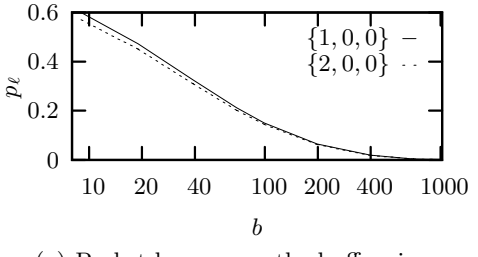


(c) Packet waiting time versus the buffer size

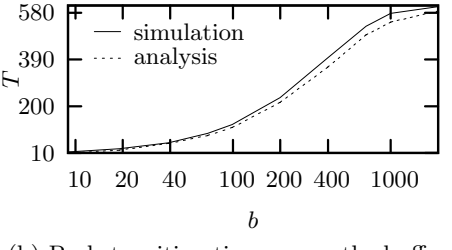


(d) Packet waiting time versus the switch size

Fig. 6. Numerical results for the packet loss analysis of LB switches



(a) Packet loss versus the buffer size



(b) Packet waiting time versus the buffer size

Fig. 7. Behavior of switches with large buffers

Part 2. Figure 7(a) and 7(b) shows the applicability of the analytical model for large buffer sizes. According to presented results, we admit that the ratio between the switch size and buffer length of the VOQs is a crucial issue for the expected packet loss and system performance. Unfortunately, the optimal set of parameters (e.g. switch size and buffer length) is not constant and should be chosen to the specific needs.

6 Conclusions

In this paper we present a scalable model for the packet loss and packet waiting time analysis in the load-balancing Birkhof-von Neumann switch.

This model also reflex the previously shown property of different loss probabilities on the chosen path traversing the switch [5,6].

The computational complexity of the approximate model introduced in this paper is reduced to be linear with N, the number of ports of the switch. The other contribution of the paper is the folding algorithm based, numerically stable and fast algorithm to solve the DTMCs for large buffer sizes (b). This allow us to solve switches of size up to ~ 30 equipped with buffer of size up to ~ 1000.

References

1. Chang, C., Lee, D., Jou, Y.: Load-Balanced Birkhoff-von Neumann switches, Part I: One-Stage Buffering. Computer Communications 25, 611–622 (2002)
2. Keslassy, I., Chuang, S., Yu, K., Miller, D., Horowitz, M., Solgaad, O., McKeown, N.: Scaling Internet Routers Using Optics. In: SIGCOMM 2003, Germany (2003)
3. Tu, C., Chang, C., Lee, D., Chiu, C.: Design a Simple and High Performance Switch Using a Two-Stage Architecture. In: IEEE GLOBECOM 2005, St. Louis, MO, USA, November 2005, vol. 2, pp. 6–11 (2005)
4. Audzevich, Y., Ofek, Y., Telek, M., Yener, B.: Analysis of load-balanced switch with finite buffers. In: IEEE Globecom 2008, New Orleans, LA, USA, pp. 1–6 (2008)
5. Audzevich, Y., Bodrog, L., Telek, M., Ofek, Y., Yener, B.: Variable Size Packets Analysis in Load-balanced Switch with Finite Buffers. In: submitted for revision to IEEE HPSR 2009 (January 2009), http://webspn.hit.bme.hu/~bodrog/techrep/hpsr2009.pdf
6. Audzevich, Y., Bodrog, L., Ofek, Y., Telek, M.: Packet Loss Analysis of Load-Balancing Switch with ON/OFF Input Processes. In: submitted for revision to EPEW 2009 (February 2009), http://webspn.hit.bme.hu/~bodrog/techrep/epew2009.pdf
7. Thompson, K., Miller, G., Wilder, R.: Wide-area Internet Traffic Patterns and Characteristics. IEEE Network 11(6), 10–23 (1997)
8. Fomenkov, M., Keys, K., Moore, D., Claffy, K.: Longitudinal study of internet traffic in 1998-2003. In: Proceedings of WISICT, Mexico, January 5-8 (2004)
9. Turner, J.: Strong Performance Guarantees for Asynchronous Crossbar Schedulers. In: IEEE INFOCOM 2006, Barcelona, Spain, April 2006, pp. 1–11 (2006)
10. Turner, J.: Resilient Cell Resequencing in Terabit Routers. Technical report, Washington University, Department of Computer Science (June 2003)
11. Ye, J., Li, S.: Courier dover publication. Folding Algorithm: A Computational Method for Finite QBD Processes with Level-Dependent Transitions 42(2/3/4), 639–652 (1994)

Mixed Finite-/Infinite-Capacity Priority Queue with General Class-1 Service Times

Thomas Demoor, Joris Walraevens, Dieter Fiems, Stijn De Vuyst, and Herwig Bruneel

Department of Telecommunications and Information Processing,
Ghent University, St.-Pietersnieuwstraat 41, B-9000 Gent, Belgium
Tel.: 003292648902; Fax: 003292644295
{thdemoor,jw,df,sdv,hb}@telin.ugent.be

Abstract. This paper studies a single-server queue with two traffic classes in order to model Expedited Forwarding Per-Hop Behaviour in the Differentiated Services architecture. Generally, queueing models assume infinite queue capacity but in a DiffServ router the capacity for high priority traffic is often small to prevent this traffic from monopolizing the output link and hence causing starvation of other traffic. The presented model takes the exact (finite) high-priority queue capacity into account. Analytical formulas for system contents and packet delay of each traffic class are determined. This requires extensive use of the spectral decomposition theorem as the service time of a high-priority packet takes a general distribution, which complicates the analysis. Numerical examples indicate the considerable impact of the finite capacity on the system performance.

Keywords: Queueing Systems and Networks, Performance Modelling.

1 Introduction

In the nodes (routers, etc.) of computer networks, packets typically have to wait before being transmitted to the next node and queues are present in order to preserve waiting packets. Roughly two types of packets can be distinguished. Real-time traffic, such as Voice-over-IP, requires low delays but can endure a small amount of packet loss. Data traffic, such as file transfer, benefits from low packet loss but has less stringent delay characteristics.

Evidently, configuring the queue in order to allow both classes to meet their Quality of Service (QoS) requirements is of paramount importance. This is enabled by implementing Differentiated Services (DiffServ), a computer networking architecture in Internet Protocol (IP) networks that classifies packets [1]. It provides QoS differentiation between traffic classes by basing the order in which packets are transmitted on class-dependent priority rules. DiffServ defines the packet forwarding properties associated with a class of traffic by using Per-Hop Behaviors (PHBs). Obviously, implementation of DiffServ is particularly interesting in wireless networks and access networks, as these typically struggle to provide acceptable QoS because bandwidth is limited and/or variable.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 264–278, 2009.

This paper considers a two-class priority queueing system representing a Diff-Serv implementation where real-time traffic (Expedited Forwarding PHB) has strict priority scheduling over data traffic (Default PHB). This is the most drastic scheduling algorithm, as data packets are only served if there are no real-time packets in the system. It thus minimizes the delay of the real-time packets. However, caution is required as these packets could occupy the server (almost) permanently, causing starvation of data traffic. This should be alleviated by controlling the amount of real-time traffic allowed into the system. Moreover, queueing a very large amount of real-time packets is useless anyway as they require small delays. These two observations emphasize the importance of limiting the capacity for real-time packets, without neglecting the packet loss constraints for these packets. On the other hand, the loss-sensitivity of data packets yields a capacity as large as practically feasible for these packets. Therefore, we can assume that the capacity for data packets is sufficiently large to be approximated by infinity but that the capacity for real-time packets should be modelled exactly. In the literature, priority queues have been discussed with various arrival and service processes. Analytic studies of queueing systems often assume infinite queue capacity facilitating mathematical analysis of the system.

From the former paragraph it indeed follows that we can assume that the capacity for data packets is sufficiently large to be approximated by infinity but that the capacity for real-time packets should be modelled as a finite number. The presented model is related to [2] where both queues are presumed to have infinite capacity and it is an extension of [3], where service of a packet was deterministically equal to a single slot for both classes. The current contribution introduces differentiation amongst packet sizes of both classes as the service time of a real-time packet takes a general distribution. This nontrivial extension leads to extensive use of the spectral decomposition theorem [4] in order to study the performance of our system. Finite queue capacity is considered in [5] as well, albeit by a different methodology, but only packet loss is investigated profoundly and delay is not analyzed at all. Assessing the impact of the finite real-time queue capacity is the main purpose of this contribution, as well as studying the effect of the general service times for real-time packets.

The remainder of this paper is organized as follows: first the model under consideration will be thoroughly described. In section 3, several performance measures for our system are determined analytically. Afterwards, the results are investigated in some (numerical) examples. The paper is concluded in section 5.

2 Model

This paper studies a discrete-time single-server two-class priority queueing system where class-1 (real-time) packets receive strict priority over class-2 (data) packets. Packets are handled in a First-In-First-Out (FIFO) manner within a class. We limit the capacity of the class-1 queue to N packets such that real-time packets that arrive at a full queue are dropped by the system. The system can hence contain up to $N + 1$ class-1 packets simultaneously, N in the queue

and 1 in the server. In contrast, the class-2 queue has infinite capacity. Time is divided into fixed-length slots and a packet can only enter the server at slot boundaries, even if arriving in an empty system.

Let s_i denote a generic random service time of a class-1 packet. Service of a class-2 packet takes a single slot (for convenience purposes), whereas service of a class-1 packet follows a general distribution with pgf $S(z)$ and mean value μ. When observing the system at the beginning of a slot this is after possible departures in the previous slot and before arrivals in the current slot.

We assume that for both classes the number of arrivals in consecutive slots form a sequence of independent and identically distributed (i.i.d.) random variables. We define $a_{i,k}$ as the number of class-i ($i = 1, 2$) packet arrivals during slot k. The arrivals of both classes are characterized by the joint probability mass function (pmf) $a(m, n) = \Pr[a_{1,k} = m, a_{2,k} = n]$ which allows us to take into account dependence between both classes. The partial probability generating function (pgf) of the number of class-2 arrivals in a slot with i ($0 \leq i \leq N$) and i or more class-1 arrivals are respectively denoted by $A_i(z)$ and $A_i^*(z)$. We establish

$$A_i(z) = \mathrm{E}[z^{a_{2,k}}\ 1\{a_{1,k} = i\}] = \sum_{j=0}^{\infty} a(i, j)z^j\ ,\ A_i^*(z) = \sum_{j=i}^{\infty} A_j(z)\ . \tag{1}$$

The indicator function $1\{.\}$ evaluates to 1 if its argument is true and to 0 if it is false. The mean number of class-1 and class-2 arrivals per slot are respectively expressed as

$$\bar{a}_1 = \sum_{i=1}^{\infty} iA_i(1)\ ,\ \bar{a}_2 = \frac{d}{dz}A_0^*(z)\bigg|_{z=1} = A_0^{*\prime}(1)\ . \tag{2}$$

The mean number of total arrivals is represented by $\bar{a}_T = \bar{a}_1 + \bar{a}_2$. Therefore, the arrival load is described as $\rho_T = \bar{a}_1\mu + \bar{a}_2$.

3 Analysis

First, we review the spectral decomposition theorem for non-diagonalisable matrices as it will be used frequently in the remainder of this paper. The next subsection addresses the characterization of arrivals during a class-1 service. The system contents are obtained at so-called start-slots and non start-slots consecutively enabling identification of the system contents at the beginning of random slots. Finally, the packet delay is obtained for both classes.

3.1 Spectral Decomposition of Non-diagonalisable Matrices

Consider a square $m \times m$ matrix $\mathbf{A}$ and a scalar function f. The spectral decomposition theorem allows us to express the image of $\mathbf{A}$ under f by evaluating f (and its derivatives) in the eigenvalues of $\mathbf{A}$, see e.g. [4].

In this paper, the function f is typically a power series $f(z) = \sum_{n=0}^{\infty} f_n z^n$ and the matrix $\mathbf{A}$ is non-diagonalisable. Such a matrix $\mathbf{A}$ cannot be reduced to a completely diagonal form by a similarity transform. However, any square matrix can be reduced to a form that is almost diagonal, called the Jordan normal form $\mathbf{J}$. Based on this reduction, it is possible to prove that the matrix $f(\mathbf{A})$ can be uniquely defined as

$$f(\mathbf{A}) = \sum_{j=1}^{s} \sum_{i=0}^{k_j-1} \frac{1}{i!} f^{(i)}(\lambda_j) \, (\mathbf{A} - \lambda_j \mathbf{I})^i \, \mathbf{G}_j \, . \tag{3}$$

In this expression, $\{\lambda_1, \ldots, \lambda_s\}$ $(s \leqslant m)$ are the eigenvalues of $\mathbf{A}$, k_j denotes the index of eigenvalue λ_j and $f^{(i)}$ is the ith derivative of f. Obviously, it is required that the function f and its derivatives exist in the eigenvalues, i.e.

$$\lambda_j \in \text{dom } f^{(i)} \, , \qquad j = 1, \ldots, s \, , i = 0, \ldots, k_j - 1 \, . \tag{4}$$

The matrices $\mathbf{G}_j$ are called the constituents or spectral projectors of $\mathbf{A}$ belonging to the eigenvalue λ_j and have the following properties:

- $\mathbf{G}_j$ is idempotent, i.e. $\mathbf{G}_j^2 = \mathbf{G}_j$.
- $\mathbf{G}_1 + \mathbf{G}_2 + \ldots + \mathbf{G}_s = \mathbf{I}$, with $\mathbf{I}$ the $m \times m$ identity matrix.
- $\mathbf{G}_j \mathbf{G}_{j'} = \mathbf{0}$ whenever $j \neq j'$ $(1 \leqslant j, j' \leqslant s)$.

In general, the matrices $\mathbf{G}_j$ need to be calculated from the transformation matrix $\mathbf{P}$, for which $\mathbf{J} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$. Specifically, if $\mathbf{P}$ is partitioned conformably as

$$\mathbf{A} = \mathbf{P}\mathbf{J}\mathbf{P}^{-1} = \begin{bmatrix} \mathbf{P}_1 \, \mathbf{P}_2 \cdots \mathbf{P}_s \end{bmatrix} \begin{bmatrix} \mathbf{J}_1 & & & \\ & \mathbf{J}_2 & & \\ & & \ddots & \\ & & & \mathbf{J}_s \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \vdots \\ \mathbf{Q}_s \end{bmatrix} \, , \tag{5}$$

with $\mathbf{J}_j$ the Jordan segment corresponding with eigenvalue λ_j, then the projectors $\mathbf{G}_j$ are

$$\mathbf{G}_j = \mathbf{P}_j \mathbf{Q}_j \qquad (j = 1, \ldots, s) \, . \tag{6}$$

We also note that the columns of $\mathbf{P}_j$ span the space of the right eigenvectors of $\mathbf{A}$ corresponding to λ_j while the rows of $\mathbf{Q}_j$ span the space of its left eigenvectors.

This spectral decomposition theorem provides us with a very powerful tool from the computational point of view. Instead of having to evaluate the matrix power series $\sum_{n=0}^{\infty} f_n \mathbf{A}^n$ we now only need to evaluate the function f and its derivatives for scalar arguments and compute a finite number of matrix multiplications. The downside is that the eigenvalues of $\mathbf{A}$ have to be calculated, as well as the matrices $\mathbf{G}_j$. But once this is done, $f(\mathbf{A})$ can easily be calculated for any function f satisfying (4). In subsection 3.2, it will become clear that in our case these downsides are virtually non-existent as the eigenvalues and spectral projectors are surprisingly easy to obtain.

3.2 Arrivals During a Class-1 Service

Let $e_{i,k}$ represent the number of class-i arrivals during a class-1 service that starts in slot k. We have

$$e_{i,k} = \sum_{m=0}^{s_1-1} a_{i,k+m} \ . \tag{7}$$

Notice that the $e_{i,k}$ are i.i.d. as the $a_{i,k}$ are i.i.d. and independent of s_1. The partial pgfs of the number of class-2 arrivals during a class-1 service, during which i $(0 \le i \le N)$ and i or more class-1 packets arrive are respectively denoted by $E_i(z)$ and $E_i^*(z)$. We have

$$E_i(z) = \mathrm{E}[z^{e_{2,k}} \ 1\{e_{1,k} = i\}] \ , \ E_i^*(z) = \sum_{m=i}^{\infty} E_m(z) \ . \tag{8}$$

Obtaining these partial pgfs can be a tedious task. During each slot of a class-1 service, packets are added to the queue according to the $(N+1)\times(N+1)$ matrix

$$\mathbf{Y}(z) = \begin{bmatrix} A_0(z) & A_1(z) & \cdots & A_{N-1}(z) & A_N^*(z) \\ 0 & A_0(z) & \cdots & A_{N-2}(z) & A_{N-1}^*(z) \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & A_0(z) & \vdots \\ 0 & \cdots & \cdots & 0 & A_0^*(z) \end{bmatrix} \ . \tag{9}$$

More precisely, given that the class-1 queue content (excluding the server) is $i-1$ during the previous slot, $\mathbf{Y}(1)_{ij}$ is the probability that it is $j-1$ in the current slot (this is the probability that $j-i$ class-1 packets are effectively allowed into the system), while $\mathbf{Y}(z)_{ij}$ is the partial pgf of the packets added to the class-2 queue.

The partial pgfs $E_i(z)$ and $E_i^*(z)$ are found as elements of the matrix $S\big(\mathbf{Y}(z)\big)$, which plays a crucial role. Using spectral decomposition, the latter is easily evaluated because of the special eigenstructure of $\mathbf{Y}(z)$. As this matrix has a triangular form, the eigenvalues simply are its diagonal elements. There are two distinct eigenvalues: $\lambda_1 = A_0^*(z)$, with index 1, and $\lambda_2 = A_0(z)$, with index N. The corresponding spectral projectors are shown to be independent of z and given by

$$\mathbf{G}_1 = \begin{bmatrix} \mathbf{0} \cdots \mathbf{0} \ \mathbf{e} \end{bmatrix} \ , \ \mathbf{G}_2 = \begin{bmatrix} \mathbf{I} & -\mathbf{e} \\ \mathbf{0}^T & 0 \end{bmatrix} \ . \tag{10}$$

Here $\mathbf{I}$ denotes the identity matrix of appropriate size, $\mathbf{x}^T$ is the transpose of vector $\mathbf{x}$ and $\mathbf{e}$ and $\mathbf{0}$ indicate the column vector of appropriate size with all elements equal to 1 and 0 respectively.

Spectral decomposition (3) yields

$$S\big(\mathbf{Y}(z)\big) = S\big(A_0^*(z)\big)\mathbf{G}_1 + \sum_{j=0}^{N-1} \frac{S^{(j)}\big(A_0(z)\big)}{j!} \big(\mathbf{Y}(z) - A_0(z)\mathbf{I}\big)^j \mathbf{G}_2 \ . \tag{11}$$

3.3 System Contents at the Beginning of Start-Slots

A start-slot is a slot where service of a packet can start. Note that a slot where the system is empty at the beginning of the slot is a start-slot as well. The class-i system contents at the beginning of start-slot l are denoted by $n_{i,l}$. The partial pgf of the class-2 system contents at the beginning of start-slot l that has class-1 system contents equal to i is denoted as

$$N_{i,l}(z) = \mathrm{E}[z^{n_{2,l}}\ 1\{n_{1,l} = i\}]\ . \tag{12}$$

The set $\{(n_{1,l}, n_{2,l}), l \geq 1\}$ forms a Markov chain. Assume that start-slot l corresponds with slot k. Relating start-slots l and $l+1$ establishes the following set of system equations:

$$\begin{aligned} n_{1,l+1} &= \begin{cases} \min(N, a_{1,k}) & \text{if } n_{1,l} = 0 \\ \min(N, n_{1,l} - 1 + e_{1,k}) & \text{if } n_{1,l} > 0 \end{cases}, \\ n_{2,l+1} &= \begin{cases} (n_{2,l} - 1)^+ + a_{2,k} & \text{if } n_{1,l} = 0 \\ n_{2,l} + e_{2,k} & \text{if } n_{1,l} > 0 \end{cases}. \end{aligned} \tag{13}$$

Here $(.)^+$ is shorthand for $\max(0, .)$. The system equations can be explained as follows: if $n_{1,l} > 0$, a class-1 packet starts service at the beginning of start-slot l and it leaves the system immediately before start-slot $l+1$. For each class, admitted arrivals during this service contribute to the system contents at the beginning of start-slot $l+1$. On the other hand, if $n_{1,l} = 0$, a class-2 packet starts service at the beginning of start-slot l if there are class-2 packets present in the system. As this service only takes a single slot, start-slot $l+1$ is the next slot. If the system is empty, the server is idle and start-slot $l+1$ is the next slot. Note that the class-1 system contents at the beginning of start-slots cannot exceed N.

We now define the $(N+1) \times (N+1)$ matrix

$$\mathbf{X}(z) = \begin{bmatrix} A_0(z) & A_1(z) & \cdots & A_{N-1}(z) & A_N^*(z) \\ E_0(z) & E_1(z) & \cdots & E_{N-1}(z) & E_N^*(z) \\ 0 & E_0(z) & \cdots & E_{N-2}(z) & E_{N-1}^*(z) \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & E_0(z) & E_1^*(z) \end{bmatrix}, \tag{14}$$

and the row vector of $N+1$ elements

$$\mathbf{n}_l(z) = \begin{bmatrix} N_{0,l}(z)\ N_{1,l}(z) \cdots\ N_{N,l}(z) \end{bmatrix}\ , \tag{15}$$

which corresponds with the system contents at the lth start-slot and we will use this phrase to determine vectors like (15) throughout this paper. Using standard z-transform techniques, a relation between $\mathbf{n}_l(z)$ and $\mathbf{n}_{l+1}(z)$ is derived from the system equations (13). We have

$$\mathbf{n}_{l+1}(z) = \mathbf{n}_l(z) \begin{bmatrix} \frac{1}{z} & & \\ & 1 & \\ & & \ddots \\ & & & 1 \end{bmatrix} \mathbf{X}(z) + \mathbf{n}_l(0) \begin{bmatrix} \frac{z-1}{z} & & \\ & 0 & \\ & & \ddots \\ & & & 0 \end{bmatrix} \mathbf{X}(z) \ . \tag{16}$$

Assume that the system has reached steady-state and define following steady-state values

$$\mathbf{n}(z) = \lim_{l\to\infty} \mathbf{n}_l(z) = \lim_{l\to\infty} \mathbf{n}_{l+1}(z) = \begin{bmatrix} N_0(z) & N_1(z) & \cdots & N_N(z) \end{bmatrix} \ . \tag{17}$$

Taking the limit of (16) for $l \to \infty$ induces

$$\mathbf{n}(z) \left(z\mathbf{I} - \begin{bmatrix} 1 & & \\ & z & \\ & & \ddots \\ & & & z \end{bmatrix} \mathbf{X}(z) \right) = \left((z-1) N_0(0) \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \mathbf{X}(z) \right) \ . \tag{18}$$

The constant $N_0(0)$ is still unknown. Note that $\mathbf{X}(1)$ is a right-stochastic matrix by construction. Therefore, observe that

$$\big(\mathbf{I} - \mathbf{X}(1)\big)\mathbf{e} = \mathbf{0} \ , \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \mathbf{X}(1)\mathbf{e} = 1 \ . \tag{19}$$

Keeping these identities in mind, derivation of (18) with respect to z, evaluation in $z = 1$ and multiplication of both sides of the resulting equation by $\mathbf{e}$ yields

$$N_0(0) = \mathbf{n}(1) \left(\mathbf{I} - \begin{bmatrix} 0 & & \\ & 1 & \\ & & \ddots \\ & & & 1 \end{bmatrix} \mathbf{X}(1) - \mathbf{X}'(1) \right) \mathbf{e} \ . \tag{20}$$

The vector $\mathbf{n}(1)$ is yet to be obtained. Evaluating (18) in $z = 1$ produces

$$\mathbf{n}(1) \Big(\mathbf{I} - \mathbf{X}(1) \Big) = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} \ . \tag{21}$$

As $\mathbf{X}(1)$ is right-stochastic, each row of matrix $[\mathbf{I} - \mathbf{X}(1)]$ sums to 0 and it hence has rank N and is not invertible. We thus require an additional relation in order to obtain the vector $\mathbf{n}(1)$. Observe that $N_i(1)$ represents the probability that the class-1 system contents at the beginning of a start-slot in steady state equal i and thus

$$N_i(1) = \lim_{l\to\infty} \Pr[n_{1,l} = i] \ . \tag{22}$$

The normalization condition provides $\mathbf{n}(1)\mathbf{e} = 1$. Combining this observation with (21) yields

$$\mathbf{n}(1) = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} \Big(\big[\mathbf{I} - \mathbf{X}(1) \big\| \mathbf{e} \big] \Big)^{-1} \ . \tag{23}$$

By $[\mathbf{A}\|\mathbf{b}]$ we denote the matrix $\mathbf{A}$ with the last column replaced by the column vector $\mathbf{b}$.

The probability mass function (pmf) of the class-1 system contents at the beginning of a start-slot in steady state has been obtained in (23). Substituting it in (20) produces $N_0(0)$, the only unknown in (18). The latter yields the pgf of the class-2 system contents at the beginning of a start-slot in steady state as

$$\lim_{l\to\infty} \mathrm{E}[z^{n_{2,l}}] = \mathbf{n}(z)\mathbf{e} \ . \tag{24}$$

3.4 Queue Contents at the Beginning of Non Start-Slot Slots

If a random slot k is not a start-slot, a class-1 packet started service in the start-slot preceding the random slot (start-slot l). We know that no packets leave the server between these two slots. Hence, we study the queue contents, instead of the system contents, at the beginning of slots that are not start-slots. The system certainly contains class-1 packets at the beginning of start-slot l, one of which enters the server (leaves the queue) at the beginning of start-slot l. Therefore, the steady-state queue contents of both classes, at the beginning of a start-slot in steady state where a class-1 packet starts service, are characterized by the vector of $N+1$ elements

$$\mathbf{m}(z) = \frac{1}{1-N_0(1)} \left[N_1(z) \cdots \ N_N(z) \ 0 \right] \ . \tag{25}$$

Slot k lies in the time epoch between start-slots l and $l+1$. No packets leave the system (and hence the queue) during this epoch. In start-slot l and in the slots up to start-slot $l+1$, packets (of both classes) arrive at the queue according to the matrix $\mathbf{Y}(z)$ given in (9). Slot k is one of the s_1-1 slots between start-slot l and $l+1$ with s_1 the service time of the class-1 packet in service. Standard renewal theory [6] yields that $\mathbf{q}(z)$, the vector of $N+1$ elements representing the queue contents of both classes at the beginning of a non start-slot in steady state is given by

$$\mathbf{q}(z) = \mathbf{m}(z) \frac{E\left[\sum_{i=1}^{s_1-1} \mathbf{Y}(z)^i\right]}{\mu - 1} \ . \tag{26}$$

Define the function $S^n(x)$ as

$$S^n(x) = E\left[\sum_{i=1}^{s_1-1} x^i\right] = \frac{S(x)-x}{x-1}. \tag{27}$$

By combining (26) and (27) and keeping in mind that the spectral decomposition theorem (3) enables evaluation of $S^n\big(\mathbf{Y}(z)\big)$, we have

$$\mathbf{q}(z) = \mathbf{m}(z) \frac{S^n\big(\mathbf{Y}(z)\big)}{\mu - 1} \ . \tag{28}$$

3.5 System Contents at the Beginning of a Random Slot

On average, a start-slot corresponds with μ slots if a class-1 packet starts service and with one slot if this is not the case (the system is void of class-1 packets). Therefore, γ, the (long-run) probability that a random slot is a start-slot, is defined as

$$\gamma = \lim_{k\to\infty} \Pr[\text{slot k is a start-slot}] = \frac{1}{N_0(1) + \big(1 - N_0(1)\big)\mu} \ . \tag{29}$$

The class-i system contents at the beginning of a random slot are denoted by $u_{i,k}$. Note that $0 \leq u_{1,k} \leq N+1$. The system contents (of both classes) at the beginning of a random slot in steady state are determined by $\mathbf{u}(z)$, a vector of $N+2$ elements. The class-1 system contents at the beginning of a start-slot never exceed N and the server contains a class-1 packet during non start-slots, yielding

$$\mathbf{u}(z) = \big[U_0(z) \cdots U_{N+1}(z)\big] = \gamma \left[\mathbf{n}(z)\ 0\right] + (1-\gamma)\left[0\ \mathbf{q}(z)\right] \ . \tag{30}$$

The pmf of the class-1 and the pgf of the class-2 system contents at the beginning of a slot are respectively determined by $\mathbf{u}(1)$ and $\mathbf{u}(z)\mathbf{e}$.

The number of class-1 packets effectively entering the system and leaving the system in steady-state must be equal. This allows us to determine the effective class-1 load ρ_1^e, the mean number of effective class-1 arrivals $\bar{a}_1^e$ and the class-1 packet loss ratio plr_1, the fraction of class-1 packets rejected by the system. We have

$$\rho_1^e = 1 - U_0(1) \ , \ \bar{a}_1^e = \frac{\rho_1^e}{\mu} \ , \ plr_1 = \frac{\bar{a}_1 - \bar{a}_1^e}{\bar{a}_1} \ . \tag{31}$$

3.6 Class-1 Delay

Tag an arbitrary class-1 packet that effectively arrives at the system in a slot in steady-state. The arrival slot of the packet is assumed to be slot k. Let the delay of the packet be denoted by d_1. Recall that class-1 packets are not affected by class-2 packets. We obtain the amount of class-1 packets in the system at the moment the tagged packet arrives. As the service times are i.i.d., each of these packets (except the class-1 packet in service during slot k) will contribute a random number of s_1 slots to the delay, as will the tagged packet itself. Therefore, once a class-1 packet arrives at the system, its delay is known.

Let $f_{1,k}$ denote the amount of class-1 packets arriving in slot k but before the tagged packet. Renewal theory states that a random packet is more likely to arrive in a slot with a lot of arrivals. This yields, considering that the tagged packet has to be an effective arrival,

$$\begin{aligned} &\Pr[f_{1,k} = m \mid (u_{1,k}-1)^+ = i] = \frac{A^*_{m+1}(1)}{\bar{a}_1^e} \ , \quad m = 0 \ldots N-i-1 \ , \\ &\Pr[f_{1,k} = m \mid (u_{1,k}-1)^+ = i] = 0 \ , \quad m > N-i-1 \ . \end{aligned} \tag{32}$$

Define the matrix $\mathbf{F}_1^e$ such that the element on row i, column j ($1 \le i \le N+1,\ 1 \le j \le N$) corresponds with $\Pr[f_{1,k} = j - i \mid (u_{1,k}-1)^+ = i-1]$. We have

$$\mathbf{F}_1^e = \frac{1}{\bar{a}_1^e} \begin{bmatrix} A_1^*(1) & A_2^*(1) & \cdots & A_N^*(1) \\ 0 & A_1^*(1) & \cdots & A_{N-1}^*(1) \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & A_1^*(1) \\ 0 & \cdots & \cdots & 0 \end{bmatrix} . \tag{33}$$

Note that the queue cannot be entirely full upon arrival of the tagged packet as the latter must be able to enter the system.

If the system does not contain class-1 packets at the beginning of slot k the delay is rather straightforward as only the tagged packet and the packets arriving before it in slot k contribute to the delay. On the other hand, if $u_{1,k} > 0$ a class-1 packet is in service and additional random variables are involved. Let s_1^- denote the elapsed service time and let s_1^+ denote the remaining service time (slot k excluded). The packet in service only contributes s_1^+ slots to the tagged packet's delay. The class-1 packets in the queue at the moment the tagged packet arrives each contribute s_1 slots to the delay. They are constituted by $\mathbf{m}(1)$, the queue content at the start-slot preceding slot k, obtained in (25), the number of arriving class-1 packets during s_1^- and $f_{1,k}$, the number of class-1 packets arriving before the tagged packet in slot k.

Define the function

$$S^b(x,y,z) \triangleq \mathrm{E}[x^{s_1^-}\, y\, z^{s_1^+}] , \tag{34}$$

where the arguments can be matrices and the order in which the arguments appear is hence important as matrix multiplication does not commute. From the discussion above follows that we need to calculate $S^b\big(\mathbf{Y}(1), \mathbf{F}_1^e, z\big)$ in order to obtain the class-1 delay. Considering that $\mathbf{F}_1^e$ does not contain stochastic variables and that scalar multiplication of a matrix commutes, we have $S^b\big(\mathbf{Y}(1), \mathbf{F}_1^e, z\big) = S^b\big(\mathbf{Y}(1), 1, z\big)\mathbf{F}_1^e$.

The random variables s_1^- and s_1^+ are generally dependent. Slot k may be any slot in s_1 with equal probability [6]. For scalar arguments x, y, z this yields

$$S^b(x,y,z) = \mathrm{E}[x^{s_1^-}\, y\, z^{s_1^+}] = \frac{S(x) - S(z)}{\mu(x-z)} y . \tag{35}$$

Using the spectral decomposition theorem (3), we can express the image of a matrix under the function S^b, as it can be seen as a scalar function in a single variable by considering the other two variables to be constant. This allows us to obtain $S^b\big(\mathbf{Y}(1), 1, z\big)$ from (35). Bringing everything together, the pgf $D_1(z)$ of the steady-state class-1 delay is given by

$$D_1(z) = \Big(\big[U_0(1)\ 0 \cdots\ 0\big] + \big(1 - U_0(1)\big)\mathbf{m}(1) S^b\big(\mathbf{Y}(1), 1, z\big) \Big) \mathbf{F}_1^e \begin{bmatrix} S(z) \\ S(z)^2 \\ \vdots \\ S(z)^N \end{bmatrix} . \tag{36}$$

3.7 Class-2 Delay

The delay of class-2 packets is more intricate as it is influenced by class-1 packets arriving at the system until the class-2 packet enters the server. In order to capture this influence we first study the (remaining) class-1 busy period.

The remaining class-1 busy period in start-slot l, denoted by r_l, is the number of slots until the system is void of class-1 packets (for the first time). Obviously, it depends on the number of class-1 packets in the system at start-slot l. The conditional pgf of the remaining class-1 busy period in start-slot l, if the class-1 system contents at the beginning of start-slot l equal j is denoted by

$$R_l(z|j) = \mathrm{E}[z^{r_l}|n_{1,l} = j],\ j = 0 \ldots N\ . \tag{37}$$

Define the vector $\mathbf{R}_l(z) = \big[R_l(z|0) \cdots R_l(z|N)\big]^T$. Relating start-slot l and $l+1$ yields

$$\mathbf{R}_l(z) = \big[1\ 0 \cdots\ 0\big]^T + \begin{bmatrix} \mathbf{0}^T & 0 \\ \mathbf{I} & \mathbf{0} \end{bmatrix} S\big(\mathbf{Y}(1)z\big)\mathbf{R}_{l+1}(z)\ . \tag{38}$$

The first term results from $R_l(z|0)$ marking the end of the (remaining) busy period as the system is empty. The second term expresses that for $R_l(z|j),\ j > 0$ the packet in service leaves the system by the next start-slot and that we keep track of the number of slots during the epoch s_1 between start-slots l and $l+1$ and the arrivals during this epoch. In each slot of this epoch class-1 packets arrive according to $\mathbf{Y}(1)$. Spectral decomposition (3) again yields evaluation of $S\big(\mathbf{Y}(1)z\big)$. In steady-state, taking the limit for l of (38) results in a simple expression for $\mathbf{R}(z) = \lim_{l\to\infty} \mathbf{R}_l(z) = \lim_{l\to\infty} \mathbf{R}_{l+1}(z)$.

A class-1 busy period b is the number of consecutive slots with class-1 system contents greater than zero. Notice that a class-1 busy period is simply the remaining class-1 busy period in a random start-slot preceded by a start-slot with empty class-1 system contents at the beginning of the slot and a number of class-1 arrivals larger than 0. Thus we obtain the pgf of the steady-state class-1 busy period as

$$B(z) = \frac{\sum_{m=1}^{N-1} R(z|m)A_m(1) + R(z|N)A_N^*(1)}{1 - A_0(1)}\ . \tag{39}$$

The extended service completion time of a class-2 packet, denoted by t_2, starts at the slot where the packet starts service and lasts until the next slot wherein a class-2 packet can be serviced [7]. If no class-1 packets arrive during the service-slot of the packet, the server can handle another class-2 packet in the next slot. If there are class-1 arrivals, we have to wait for a class-1 busy period after the service-slot until the service of another class-2 packet can start. We can thus express the pgf of the extended service completion time in steady state as $T_2(z) = A_0(1)z + \big(1 - A_0(1)\big)B(z)z$.

Now, we can finally tackle the class-2 delay. Tag an arbitrary class-2 packet arriving at the system in a slot in steady-state. The arrival slot of the packet is assumed to be slot k. Let the delay of the packet be denoted by d_2. It resembles

the class-1 delay but here we need to keep track of packets of both classes. Consider the first start-slot succeeding slot k. The remainder of the delay of the tagged packet is simply the remaining class-1 busy period in this start-slot followed by an extended service completion time for each class-2 packet to be served before the tagged packet and a single slot to serve the tagged packet itself.

Let $f_{2,k}$ denote the amount of class-2 packets arriving in slot k but before the tagged packet. We determine the number of class-2 arrivals before the tagged packet. It is clear that $a_{1,k}$ and $f_{2,k}$ are correlated. The corresponding matrix can be found using renewal arguments [6]. We have

$$\hat{\mathbf{A}}(z) = \frac{\mathbf{Y}(z) - \mathbf{Y}(1)}{\bar{a}_2(z-1)} \ . \tag{40}$$

Given that the class-1 queue contents are $i-1$ at the beginning of the slot, $\hat{\mathbf{A}}(z)_{ij}$ is the partial pgf of the class-2 packets arriving before the tagged packet while $j-i$ class-1 packets are effectively allowed into the system in this slot.

We obtain the system state in the first start-slot succeeding slot k as follows. If the system does not contain class-1 packets at the beginning of slot k the next start-slot is simply the next slot and the class-2 system contents at the beginning of slot k (if any) contribute to the delay. On the other hand, if $u_{1,k} > 0$ a class-1 packet is in service and additional random variables are involved. Let s_1^- denote the elapsed service time and let s_1^+ denote the remaining service time (slot k excluded). The packets contributing to the delay are $\mathbf{m}(z)$, the queue contents at the start-slot preceding slot k, obtained in (25), the number of arriving packets of both classes during s_1^- and the number of arriving class-1 packets during s_1^+. Note that s_1^+ contributes to the delay as well.

This discussion leads to the following pgf $D_2(z)$ of the steady-state class-2 delay as

$$\begin{aligned} D_2(z) = \Bigg(& \left[\frac{U_0\big(T_2(z)\big) + \big(T_2(z)-1\big)U_0(0)}{T_2(z)} \ 0 \cdots \ 0 \right] \hat{\mathbf{A}}(T_2(z)) \\ & + \big(1 - U_0(1)\big)\mathbf{m}(T_2(z)) S^b\big(\mathbf{Y}(T_2(z)), \hat{\mathbf{A}}(T_2(z)), \mathbf{Y}(1)z\big) \Bigg) \mathbf{R}(z) z \ . \end{aligned} \tag{41}$$

Finally we calculate $S^b\big(\mathbf{Y}(T_2(z)), \hat{\mathbf{A}}(T_2(z)), \mathbf{Y}(1)z\big)$. As matrices generally do not commute, there is no multivariate version of the spectral decomposition theorem. However, if we specify the function S^b by its power series expansion we can apply the spectral decomposition theorem on the arguments separately. Power series expansion produces

$$\begin{aligned} S^b\big(\mathbf{Y}\big(T_2(z)\big), \hat{\mathbf{A}}(T_2(z)), \mathbf{Y}(1)z\big) &= E\left[\mathbf{Y}\big(T_2(z)\big)^{s_1^-} \hat{\mathbf{A}}(T_2(z)) \big(\mathbf{Y}(1)z\big)^{s_1^+} \right] \\ &= \frac{1}{\mu} \sum_{n=0}^{\infty} \text{Prob}[s_1 = n+1] \sum_{i=0}^{n} \mathbf{Y}\big(T_2(z)\big)^i \hat{\mathbf{A}}(T_2(z)) \big(\mathbf{Y}(1)z\big)^{n-i} \end{aligned} \tag{42}$$

Spectral decomposition (3) enables evaluation of $\mathbf{Y}\big(T_2(z)\big)^i$ and $\big(\mathbf{Y}(1)z\big)^{n-i}$. Note that both decompositions share the same spectral projectors $\mathbf{G}_1$ and $\mathbf{G}_2$. The eigenvalues and their index are respectively denoted by

$$\begin{aligned}\lambda_1 &= A_0^*(T_2(z)) \text{ with } k_1 = 1 \ , \ \lambda_2 = A_0(T_2(z)) \text{ with } k_2 = N \ , \\ \lambda_1' &= A_0^*(1)z \text{ with } k_1' = 1 \ , \ \lambda_2' = A_0(1)z \text{ with } k_2' = N \ .\end{aligned} \tag{43}$$

After the spectral decomposition we can reconstruct the power series yielding

$$\begin{aligned}&S^b\big(\mathbf{Y}\big(T_2(z)\big), \hat{\mathbf{A}}(T_2(z)), \mathbf{Y}(1)z\big) \\ &\qquad = \sum_{j=1}^{2} \sum_{i=0}^{k_j-1} \sum_{j'=1}^{2} \sum_{i'=0}^{k_{j'}'-1} Q_{ii'}(\lambda_j, \lambda_{j'}')(\mathbf{Y}\big(T_2(z)\big) - \lambda_j \mathbf{I})^i \mathbf{G}_j \\ &\qquad\qquad \times \hat{\mathbf{A}}(T_2(z)) \left(\mathbf{Y}(1)z - \lambda_{j'}' \mathbf{I}\right)^{i'} \mathbf{G}_{j'} \ . \\ &\text{with } Q_{ii'}(\lambda_j, \lambda_{j'}') \triangleq \frac{1}{i!}\frac{1}{i'!}\frac{\partial^{i+i'}}{\partial x^i y^{i'}} S^b\big(x, 1, y\big)\Big|_{\substack{x=\lambda_j \\ y=\lambda_{j'}'}} \ .\end{aligned} \tag{44}$$

By taking proper derivatives of the pgfs obtained in this paper, all moments of the corresponding random variables can be calculated.

4 Numerical Examples

With the formulas at hand, we study an output-queueing switch with L inlets and L outlets and two types of traffic as in [2]. On each inlet a batch arrives according to a Bernoulli process with parameter ν_T. A batch contains b (fixed) packets of class 1 with probability ν_1/ν_T or b packets of class 2 with probability ν_2/ν_T (with $\nu_1 + \nu_2 = \nu_T$). Incoming packets are routed uniformly to the outlets where they arrive at a queueing system as described in this paper. Therefore, all outlets can be considered identical and analysis of one of them is sufficient. The arrival process at the queueing system can consequently be described by the pmf

$$a(bn, bm) = \frac{L!\left(\frac{\nu_1}{L}\right)^n \left(\frac{\nu_2}{L}\right)^m \left(1 - \frac{\nu_T}{L}\right)^{L-n-m}}{n!m!(L-n-m)!}, \quad n + m \le L \ , \tag{45}$$

and by $a(p, q) = 0$, for all other values of p and q. Obviously the number of arrivals of class-1 and class-2 are negatively correlated as there can be no more than $Lb - i$ class-2 arrivals in a slot with i class-1 arrivals. For increasing values of L the correlation increases and for L going to infinity the numbers of arrivals of both types become uncorrelated. We now study a 4×4 output-queueing switch.

For Fig. 1, let $\nu_1 = \nu_2$. On average the system thus receives the same amount of packets of both classes. On the left, the batch size is $b = 10$, $\nu_1 = \nu_2 = 0.02$ and service of a class-1 packet takes the distribution $S(z) = 0.25z + 0.75z^4$ yielding a mean class-1 of service time $\mu = 3.25$ slots and hence $\rho_T = 0.85$. The mean and the standard deviation of the system contents at the beginning of random slots of both classes are plotted versus the class-1 queue capacity N. The values increase for increasing N, as the number of dropped class-1 packets decreases. For larger N the values clearly converge to the values corresponding with the

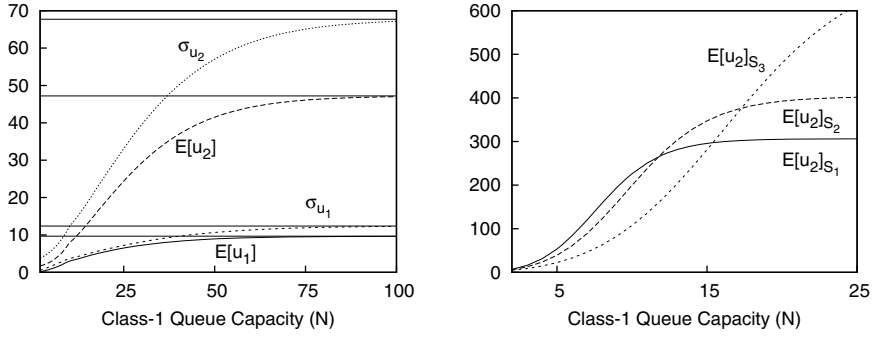


Fig. 1. System contents versus class-1 queue capacity

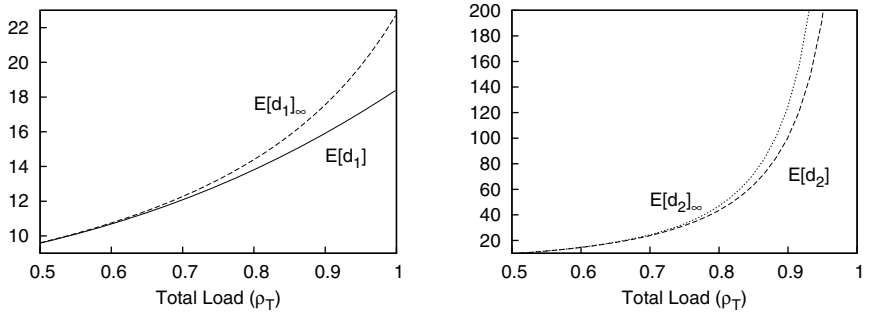


Fig. 2. Mean delays versus total load

infinite system [2], represented by the horizontal lines. However, the convergence is rather slow, especially for class-2. On the right, $b = 1$, $\nu_1 = \nu_2 = 0.199$ and we have plotted the mean class-2 system contents versus the class-1 queue capacity for three distributions with $\mu = 4$ slots yielding $\rho_T = 0.995$. These distributions have different variances. In order of increasing variance, we have

$$S_1(z) = z^4 \ , \ S_2(z) = 0.25(z + z^3 + z^5 + z^7) \ , \ S_3(z) = 0.7z + 0.3z^{11} \ . \qquad (46)$$

For large values of N the expected (from the infinite model) behaviour arises as increased variance normally yields increased system content. However, for small values of N the inverse effect occurs as the class-1 queue is likely to get full during a large service time causing arriving packets to be dropped. Therefore, the effective class-1 load will be lower when the variance of the class-1 service times is larger, increasing class-2 performance. As the queue capacity gets bigger less packets are lost and the normal behaviour is exemplified. Evidently, this effect cannot be predicted by infinite capacity queueing models.

For Fig. 2, we assume that $b = 3$, $\nu_1 = \nu_2$ and that the service of a class-1 packet takes the distribution $S(z) = 0.25z + 0.75z^4$. We keep $N = 15$ constant and vary ν_1 and ν_2 and hence the total load. The class-1 delay (on the left)

and the class-2 delay (on the right) are plotted versus the total load and are compared to results for the infinite model. We clearly see the effect of the priority scheduling as the low mean for the class-1 delay delivers the performance required for real-time traffic at the cost of the class-2 delay. Note that the starvation effect is alleviated (compared to the infinite model) when the load gets high, as an increasing amount of class-1 packets are dropped, in turn improving the delay performance of packets (of both classes) allowed into the system.

5 Conclusions

A two-class priority queue with finite capacity for high-priority packets has been studied in order to model a DiffServ router with Expedited Forwarding Per-Hop Behaviour for high-priority traffic. The service times of class-1 packets are generally distributed, which considerably complicates the analysis. Analytical formulas for system content and packet delay of all traffic classes were determined making extensive use of the spectral decomposition theorem. In a DiffServ router, the capacity for high-priority packets is often small to prevent this traffic monopolizing the system. Opposed to existing models, the presented model takes the exact (finite) high-priority queue capacity into account. The resulting impact on system performance is clearly indicated by numerical examples.

Acknowledgements. The second and third authors are postdoctoral fellows with the Research Foundation Flanders (F.W.O.-Vlaanderen), Belgium.

References

1. Carpenter, B.E., Nichols, K.: Differentiated services in the Internet. Proceedings of the IEEE 90(9), 1479–1494 (2002)
2. Walraevens, J., Steyaert, B., Bruneel, H.: Performance analysis of a single-server ATM queue with a priority scheduling. Computers & Operations Research 30(12), 1807–1829 (2003)
3. Demoor, T., Walraevens, J., Fiems, D., Bruneel, H.: Mixed finite-/infinite-capacity priority queue with interclass correlation. In: Al-Begain, K., Heindl, A., Telek, M. (eds.) ASMTA 2008. LNCS, vol. 5055, pp. 61–74. Springer, Heidelberg (2008)
4. Meyer, C.D.: Matrix Analysis and Applied Linear Algebra. Society for Industrial and Applied Mathematics, 599–615 (2000)
5. Van Velthoven, J., Van Houdt, B., Blondia, C.: The impact of buffer finiteness on the loss rate in a priority queueing system. In: Horváth, A., Telek, M. (eds.) EPEW 2006. LNCS, vol. 4054, pp. 211–225. Springer, Heidelberg (2006)
6. Takagi, H.: Queueing Analysis. Discrete-Time Systems, vol. 3. Elsevier Science Publishers, Amsterdam (1993)
7. Fiems, D.: Analysis of discrete-time queueing systems with vacations. PhD thesis. Ghent University (2003)

Stochastic Automata Networks with Master/Slave Synchronization: Product Form and Tensor

Thu Ha Dao Thi and Jean Michel Fourneau

PRiSM, Université de Versailles-Saint-Quentin, CNRS, UniverSud, Versailles, France

Abstract. We present some Continuous Time Stochastic Automata Networks (SAN) based on Master/Slave synchronizations with a product form steady-state distribution. The proof is purely algebraic and is based on some simple properties of the tensor product. The result generalizes many known theorems on product form of queueing networks.

1 Introduction

Stochastic Automata Networks (SAN for short) are since their introduction [21] associated to efficient numerical analysis of Markov chains (see for instance [2,3,6,23]). A tool have been developed [22] and the tensor representation proved for SAN has been generalized to Stochastic Petri Nets [5] and Stochastic Process Algebra [20]. This representation allows to describe the transition rate matrix of a continuous-time Markov chain as tensor products and sums for small matrices associated to the automata and more generally to the components. More formally Plateau proved in [21]:

$$Q = \bigotimes_{l=1}^{n}{}_g \, \mathbf{L_l} + \sum_{r=1}^{s} (\bigotimes_{i=1}^{n}{}_g \, \mathbf{M_i^r} + \mathbf{N^r}), \tag{1}$$

where n is the number of automata, s is the number of synchronizations, L^i and M_i^r are matrices which describe respectively the local transitions and the effect of synchronization r on automaton i. N_l^r is the normalization of M_i^r $\otimes_g$ and $\oplus_g$ denote the generalized tensor product and generalized tensor sum. These operators have been generalized to handle functional rates and probabilities in the definition of the SAN. The components interact with functional rates (i.e. rates which depend on the states of all automata) and synchronizations. Intuitively, synchronized transitions are the interactions we are used to consider in queueing networks models while functional rates are typically used in statistical physics (for instance Ising model). Here we only consider here models without functions, we only have to use a simpler version of this equation with ordinary tensor sum and product. We do not present here the general theory which now can be found in many publications [6,9,21,22,23].

Recently, the tensor representation was also associated to analytical closed-form solutions such as product form. The main idea is the simple fact that a

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 279–293, 2009.

product distribution is the tensor product of marginal distributions. Therefore as the solution and the transition rate matrix are both described by tensors products and sums, it may be possible to verify the balance equations using the tensor algebra.

This approach has been used in [8,9] to find a very simple algebraic sufficient condition for product form of steady-state solution of continuous-time SAN without synchronizations where interactions are limited to functional rates. Similarly a more complex sufficient condition has been proved in [10] for discrete-time SAN without synchronizations. These conditions are based on the existence of a common vector in the kernel of all the matrices obtained when the functional rates change. The results obtained by this approach generalize Boucherie's theory of Markov chains in competition [1].

Here we consider Stochastic Automata Networks with synchronization and without functions. Furthermore we consider the following constraints on synchronization: only two automata are involved on any synchronization. We assume that the synchronization follows the Master/Slave paradigm. The Master automata triggers the synchronization and the slave follows even if it is allowed that the slave perform a self loop. Thus the synchronization we consider are generalization of a customer or a signal movement in a generalized network of queues (usually denoted as G-networks).

In fact as we deal with tensor representation of the Markovian transition rate matrix, the high level formalism we consider is not really important. Our results only relies on algebraic properties of the matrices used to describe the components. These matrices are the inputs of the model for a SAN but they can easily be found for a Markovian network of queues or generated by a reachability algorithm for a stochastic Petri net or a Stochastic Process Algebra.

The proof is purely algebraic and we generalize many well known results such as the product form for Jackson's network or for Gelenbe's network with positive and negative customer. We also prove many new results and we give a very simple framework to found new product form results but also to build new approximations. This proof is based on algebraic properties of tensors. This type of proof was presented in [11] for a more complex synchronization denoted as a Domino. The idea developed here is to consider the simplest synchronization (i.e. the Master/Slave) to understand which properties of the tensor are really necessary to derive the proof. An extension could be to generalize the tensor approach to Zero-Automatic networks [4]. As SANs are very close to PEPA, many results proved on PEPA apply also on SAN (for instance [15,16,17,18,19]). But our objective here is to present a new type of proof which is purely algebraic while the references cited before give a probabilistic interpretation of the sufficient assumptions for product form. We want to emphasis that all the examples presented in section 4 may clearly be proved by RCAT Theorem [18].

The rest of the papers is as follows. First in section 2, we give a small introduction to the properties of tensor we need and we describe Master/Slave synchronizations and give their tensor representations. Section 3 is devoted to

the main theorem while we present in section 4 many well known results and new product form networks as well.

2 Master Slave Synchronization

In this paper, we restrict ourself to continuous-time SAN without functions. The generator is based on the tensor sum and product local components. Recall that with

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix},$$

the tensor sum $A \oplus B$ is given by:

$$\left(\begin{array}{ccc|ccc} a_{11}+b_{11} & b_{12} & b_{13} & a_{12} & 0 & 0 \\ b_{21} & a_{11}+b_{22} & b_{23} & 0 & a_{12} & 0 \\ b_{31} & b_{32} & a_{11}+b_{33} & 0 & 0 & a_{12} \\ \hline a_{21} & 0 & 0 & a_{22}+b_{11} & b_{12} & b_{13} \\ 0 & a_{21} & 0 & b_{21} & a_{22}+b_{22} & b_{23} \\ 0 & 0 & a_{21} & b_{31} & b_{32} & a_{22}+b_{33} \end{array}\right),$$

and the tensor product $A \otimes B$ is:

$$\left(\begin{array}{ccc|ccc} a_{11}b_{11} & a_{11}b_{12} & a_{11}b_{13} & a_{12}b_{11} & a_{12}b_{12} & a_{12}b_{13} \\ a_{11}b_{21} & a_{11}b_{22} & a_{11}b_{23} & a_{12}b_{21} & a_{12}b_{22} & a_{12}b_{23} \\ a_{11}b_{31} & a_{11}b_{32} & a_{11}b_{33} & a_{12}b_{31} & a_{12}b_{32} & a_{12}b_{33} \\ \hline a_{21}b_{11} & a_{21}b_{12} & a_{21}b_{13} & a_{22}b_{11} & a_{22}b_{12} & a_{22}b_{13} \\ a_{21}b_{21} & a_{21}b_{22} & a_{21}b_{23} & a_{22}b_{21} & a_{22}b_{22} & a_{22}b_{23} \\ a_{21}b_{31} & a_{21}b_{32} & a_{21}b_{33} & a_{22}b_{31} & a_{22}b_{32} & a_{22}b_{33} \end{array}\right).$$

The tensor product and sums have many algebraic properties (see [6] for proofs). We give some of them in the following for the sake of completeness. It is worthy to remark that the conditions of the theorem are based on the same kind of properties used in [8,9,10] to prove product form steady-state distributions for other types of SAN. The key property is the fact that a product form solution of n distributions $(\pi_l)_{l=1..n}$ can be written as $C\pi_1 \otimes \pi_2 \otimes \ldots \otimes \pi_n$.

Property 1 (Basic properties of Tensor Product). Let A, B and C, $A1$, $A2$, $B1$, $B2$ be arbitrary matrices, the following properties hold:

- Associativity: $(A \otimes B) \otimes C = A \otimes (B \otimes C)$.
- Distributivity over Addition:

$$(A_1 + A_2) \otimes (B_1 + B_2) = A_1 \otimes B_1 + A_1 \otimes B_2 + A_2 \otimes B_1 + A_1 \otimes B_2.$$

- Compatibility with matrix multiplication: For all vectors π_A and π_B whose sizes are consistent we have:

$$(\pi_A \otimes \pi_B)(A \otimes B) = (\pi_A A) \otimes (\pi_B B).$$

The state space of the system is the Cartesian product of the states of the automata which are combined in the network. The effective state space is in general only a subset of this product. The synchronization formerly used for SAN are defined as "Rendez-Vous". This simply says that a synchronized transition is possible, if and only if, all automata are ready for this synchronized transition. We have to consider a variant of the rendez-vous : the master-slave synchronization. As we synchronize automata and we do not allow functional rates, the generator is given by:

$$Q = \otimes_{l=1}^{n} \mathbf{L_l} + \sum_{r=1}^{s} (\otimes_{i=1}^{n} \mathbf{M_i^r} - \mathbf{N^r}), \tag{2}$$

with for all r, $\mathbf{M_i^r} = \mathbf{I}$ for all i except the two distinct indices used to describe a synchronization. More formally,

Definition 1. *Let r be a synchronization number or label. The Master/Slave synchronization consists of an ordered list of two automata called the master $msr(r)$ and the slave $sl(r)$. The master of synchronization r is the initiator of the synchronization. It performs a real transition (i.e. a loop is not allowed). The slave always follows but it is allowed to perform a loop. As the loop is not a valid transition for the master during the synchronization, the global transition of the system is not a dummy transition.*

Remark 1. Note that this definition of synchronization implies that the master is never blocked by the slave (it is not a general rendez-vous). This implies that every state of automaton $sl(r)$ is the origin of at least one synchronized transition marked by synchronization label r.

The automata are defined by the following matrices which may be either finite or infinite:

- n transition rate matrices denoted as $\mathbf{L_l}$ for automaton l. $\mathbf{L_l}$ models the rates of local transitions. The matrices are normalized, i.e.

$$\mathbf{L_l}[k,i] \geq 0 \; if \; i \neq k \;\; and \;\; \sum_i \mathbf{L_l}[k,i] = 0.$$

- s tuples of two matrices $(\mathbf{D^{(r)}}, \mathbf{E^{(r)}})$. In the tensor product associated to Master/Slave synchronization r $\otimes_{i=1}^{n} \mathbf{M_i^r}$ all matrices except $\mathbf{D^{(r)}}$ and $\mathbf{E^{(r)}}$ are equal to Identity. In the usual description of a SAN [21] the master of a synchronization is a transition rate matrix and the other matrices used in the tensor product are transition probability matrices. We use the same formulation here. In $\mathbf{D^{(r)}}$ we find the transitions due to synchronization r on the master automaton. It is assumed that the synchronizations always have an effect on the master (i.e. its transition is not a loop).

  The effect of synchronization r on the slave (i.e. automaton $sl(r)$) is specified by matrix $\mathbf{E^{(r)}}$. $\mathbf{E^{(r)}}$ is a transition probability matrix.

$$\begin{array}{l} \mathbf{D^{(r)}}[k,i] \geq 0 \; if \; i \neq k \;\; and \;\; \sum_i \mathbf{D^{(r)}}[k,i] = 0, \\ \mathbf{E^{(r)}}[k,i] \geq 0 \;\; and \;\; \sum_i \mathbf{E^{(r)}}[k,i] = 1. \end{array}$$

To complete the description of the generator of the SAN, one must give the description of the normalization associated to synchronization r. Let $\mathbf{N^r}$ be this matrix. It is a non positive diagonal matrix.

Definition 2. *Let M be a matrix, as usual $diag(M)$ is a diagonal matrix whose elements are the diagonal elements of M.*

For the sake of readability, we assume that the SAN is suitably reordered such that the automata involved in synchronization r are the first two ones. The description of the other automata is simply an Identity which is denoted here as I_1 to avoid the confusion. The SAN description associated to Master/Slave synchronization r consists in 2 terms:

1. $(\mathbf{D^{(r)}} - diag(\mathbf{D^{(r)}})) \otimes \mathbf{E^{(r)}} \otimes I_1$: synchronization.
2. $diag(\mathbf{D^{(r)}}) \otimes I \otimes I_1$: normalization of term 1.

3 Product Form Solution

We now establish a sufficient condition for a SAN with Master/Slave synchronization to have steady-state distribution which is obtained as the product of the steady-state distributions of the automata in isolation. To keep the proofs as clear as possible, we use in the following indices i, j, k and m for states, l for an automaton, r for a synchronization.

Theorem 1. *Consider a SAN with n automata and s Master/Slave synchronizations. Let $(X_1, X_2, \cdots, X_n)$ be the global state and X_l the state of component l. Assume that the continuous-time Markov chain associated with the SAN is ergodic. Consider matrices $\overline{\mathbf{D^{(r)}}} = \mathbf{D^{(r)}} - diag(\mathbf{D^{(r)}})$ and $\mathbf{E^{(r)}}$ associated to the description of synchronization r. Let g_l be such a left eigenvector of $\overline{\mathbf{D^{(r)}}}$ associated to eigenvalue Γ_r. If g_l is in the kernel of matrix $\left[\mathbf{L_l} + \sum_{r=1}^{s} \left(\mathbf{D^{(r)}} 1_{msr(r)=l} + \Gamma^{(r)}(\mathbf{E^{(r)}} - I) 1_{sl(r)=l}\right)\right]$, then the steady-state distribution has a product form solution:*

$$Pr(X_1, X_2, \cdots, X_n) = C \prod_{l=1}^{n} g_l(X_l), \tag{3}$$

and C is a normalization constant.

The proof is based on some properties of tensor products which are presented at the end of this section. Let us rewrite the conditions of the theorem: there exists a solution $(g_l)_l, (\Gamma^{(r)})_r$ to the fixed point system:

$$\begin{cases} \Gamma^{(r)}\ g_l = g_l\ \overline{\mathbf{D^{(r)}}} \quad if \quad msr(r) = l, \\ g_l\ \left[\mathbf{L_l} + \sum_{r=1}^{s} \left(\mathbf{D^{(r)}} 1_{msr(r)=l} + \Gamma^{(r)}(\mathbf{E^{(r)}} - I) 1_{sl(r)=l}\right)\right] = 0, \end{cases} \tag{4}$$

A simple interpretation may be given to these equations. The equation defines g_l as the invariant distribution (up to a normalization constant) of a continuous-time Markov chain which models the automaton in isolation (i.e. $g_l \mathbf{M_l} = 0$), with:

$$\mathbf{M_l} = \mathbf{L_l} + \sum_{r=1}^{s} \mathbf{D^{(r)}} 1_{msr(r)=l} + \sum_{r=1}^{s} \Gamma^{(r)}(\mathbf{E^{(r)}} - I) 1_{sl(r)=l}. \tag{5}$$

Remember that $\mathbf{E^{(r)}}$ is a stochastic matrix. Thus $\mathbf{E^{(r)}} - I$ is a generator. As $\mathbf{L_l}$ and $\mathbf{D^{(r)}}$ are generators, and $\Gamma^{(r)}$ and Ω_r are positive, matrix M_l is the generator of a continuous-time Markov chain. Of course, this construction does not prove in general that the chain is ergodic. However, if the chain is finite and if matrix $\mathbf{L_l}$ is irreducible, then matrix $\mathbf{M_l}$ is irreducible and the chain of the automaton in isolation is ergodic. Furthermore, the terms of the summation have an intuitive interpretation. The first term corresponds to the local transitions. The last two terms represent the effects of the synchronization on the automata. The effect on the master are explicitly represented by the transition matrix $\mathbf{D^{(r)}}$ while the effect on the slave are represented by matrix $\mathbf{E^{(r)}} - I$ multiplied by an appropriate rate. This rate is obtained from defined the first equation of the fixed point system $\Gamma^{(r)}\ g_l = g_l\ \overline{\mathbf{D^{(r)}}}$. This equation states that $\Gamma^{(r)}$ is the left-eigenvalue associated to eigenvector g_l for an operator obtained from matrix $\mathbf{D^{(r)}}$ by removing the diagonal elements. The examples presented in the next section show that this equation is a generalization of queueing networks flow equation. Note that, like in product form queueing network, the existence of these eigenvalues $\Gamma^{(r)}$ does not imply that the whole network send a Poisson streams of synchronization on automaton l. Similarly, the product form holds even if the underlying Markov chain is not reversible.

It is worthy to remark that the ergodicity of the CTMC must be assumed. We present in the next section an example where the fixed point system has a solution and the CTMC is not irreducible. Therefore the existence of a solution to the fixed point system does not imply ergodicity.

We present in the next section some examples where the product form holds. Before let us proceed with the proof of the theorem using relations between tensor products and product form distributions we have already used in [8,9,10].

3.1 Proof of the Theorem

Consider the generator or the SAN:

$$Q = \otimes_{l=1}^{n} \mathbf{L_l} + \sum_{r=1}^{s} (\otimes_{i=1}^{n} \mathbf{M_i^r} - \mathbf{N^r}), \tag{6}$$

with for all r, $\mathbf{M_i^r} = \mathbf{I}$ for all i except for the master and the slave of synchronization r. A steady-state distribution of the SAN is a probability vector π which satisfies $\pi Q = 0$. Assume that π has product form $C g_1 \otimes g_2 \otimes \ldots \otimes g_n$. Thus one must check that:

$$(g_1 \otimes g_2 \otimes \ldots \otimes g_n)(\otimes_{l=1}^{n} \mathbf{L_l}) + \sum_{r=1}^{s} (g_1 \otimes g_2 \otimes \ldots \otimes g_n)((\otimes_{i=1}^{n} \mathbf{M_i^r}) + \mathbf{N^r}) = 0. \tag{7}$$

First let us remember that $\mathbf{A} \oplus \mathbf{B} = \mathbf{A} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{B}$. Therefore the tensor sum becomes the sum of n tensor products of n matrices ($n-1$ of which are equal to Identity). We then apply the compatibility with ordinary product and we remark that $g_l \mathbf{I} = g_l$ to simplify the tensor product.

We have $n+2s$ products of n terms. The key idea is to factorize into n terms such that each term is a tensor product of n vectors. Furthermore each of these products is equal to zero because one of the vectors is zero. More precisely, each of these terms is equal to: $(g_1\mathbf{W_1} \otimes g_2\mathbf{W_2} \otimes \ldots \otimes g_n\mathbf{W_n})$ and all matrices $\mathbf{W_i}$ are equal to Identity except one which is equal to $\mathbf{M_l}$ (defined in Equation 5). As $g_l\mathbf{M_l} = 0$, the tensor product is zero.

For the sake of readability we first present the proof for the first synchronization and the automata involved in this synchronization. We also take into account the local transitions for these automata. We assume that the SAN has been reordered such that these automata the master is the first automaton and the slave is the second one. The description of $(g_1 \otimes g_2 \otimes \ldots \otimes g_n)Q$ consists in 4 terms (two coming from the tensor sum, one for the Master/Slave synchronization and one for the normalization of the synchronization):

$$\begin{aligned}
&(g_1\mathbf{L}_1 \otimes g_2 \otimes \ldots \otimes g_n)\\
+\ &(g_1 \otimes g_2\mathbf{L}_2 \otimes \ldots \otimes g_n)\\
+\ &(g_1(\mathbf{D^{(r)}} - diag(\mathbf{D^{(r)}})) \otimes g_2\mathbf{E^{(r)}} \otimes \ldots \otimes g_n)\\
+\ &(g_1 diag(\mathbf{D^{(r)}}) \otimes g_2 I \otimes \ldots \otimes g_n)
\end{aligned}$$

Now remember that $g_1(\mathbf{D^{(r)}} - diag(\mathbf{D^{(r)}})) = g_1\Gamma_r$. And of course $g_2\mathbf{I} = g_2$. After simplification, we get:

$$\begin{aligned}
&(g_1\mathbf{L}_1 \otimes g_2 \otimes \ldots \otimes g_n)\\
+\ &(g_1 \otimes g_2\mathbf{L}_2 \otimes \ldots \otimes g_n)\\
+\ &(g_1\Gamma_r \otimes g_2\mathbf{E^{(r)}} \otimes \ldots \otimes g_n)\\
+\ &(g_1 diag(\mathbf{D^{(r)}}) \otimes g_2 \otimes \ldots \otimes g_n)
\end{aligned}$$

Now, we remark that $g_1\Gamma_r \otimes g_2\mathbf{E^{(r)}} = g_1 \otimes g_2\Gamma_r\mathbf{E^{(r)}}$ because the ordinary product is compatible with the tensor product. We factorize the first and the last terms and we do the same for the second and the third term. Furthermore we add and subtract the following term: $(g_1(\mathbf{D^{(r)}} - diag(\mathbf{D^{(r)}})) \otimes g_2 \otimes \ldots \otimes g_n)$.

$$\begin{aligned}
&(g_1(\mathbf{L}_1 + diag(\mathbf{D^{(r)}})) \otimes g_2 \otimes \ldots \otimes g_n)\\
+\ &(g_1 \otimes g_2(\mathbf{L}_2 + \Gamma_r\mathbf{E^{(r)}}) \otimes \ldots \otimes g_n)\\
-\ &(g_1(\mathbf{D^{(r)}} - diag(\mathbf{D^{(r)}})) \otimes g_2 \otimes \ldots \otimes g_n)\\
+\ &(g_1(\mathbf{D^{(r)}} - diag(\mathbf{D^{(r)}})) \otimes g_2 \otimes \ldots \otimes g_n)
\end{aligned}$$

We factorize the first and the last term and we note that $g_1(\mathbf{D^{(r)}} - diag(\mathbf{D^{(r)}})) = g_1\Gamma_r$ to simplify the third term:

$$\begin{aligned}
&(g_1(\mathbf{L}_1 + \mathbf{D^{(r)}}) \otimes g_2 \otimes g_3 \otimes \ldots \otimes g_n)\\
+\ &(g_1 \otimes g_2(\mathbf{L}_2 + \Gamma_r\mathbf{E^{(r)}}) \otimes \ldots \otimes g_n)\\
-\ &(g_1\Gamma_r \otimes g_2 \otimes \ldots \otimes g_n)
\end{aligned}$$

Again we use the compatibility of the ordinary product with the tensor product and we get after factorization:

$$\begin{aligned}
&(g_1(\mathbf{L}_1 + \mathbf{D^{(r)}}) \otimes g_2 \otimes g_3 \otimes \ldots \otimes g_n)\\
+\ &(g_1 \otimes g_2(\mathbf{L}_2 + \Gamma_r(\mathbf{E^{(r)}} - I)) \otimes \ldots \otimes g_n)
\end{aligned}$$

This is the decomposition we need. Now we can continue with the second synchronization and factorize the terms to obtain n tensor products. Each of them contains a product by vector $g_l M_l$ which is zero due to the assumptions of the theorem. Therefore $(g_1 \otimes \ldots \otimes g_n)Q = 0$ and the SAN has a product form steady state distribution.

4 Examples

Before proceeding with the examples we give notation which are useful to describe the matrices involved in the models.

- **I**: the identity matrix,
- **Upp**: the matrix full of 0 except the main upper diagonal which is 1,
- **Low**: the matrix full of 0 except the main lower diagonal which is 1,
- $\mathbf{I^0}$: the identity matrix except the first diagonal element which is 0.
- **J0**: the null matrix except the first column whose elements are equal to 1.

4.1 Jackson Networks of Queues

First consider a Jackson's network of queues. Each automaton is associated to a queue. The states of one automaton is the number of customer in the associated queue. The synchronization between Automaton i and Automaton j describes the customer movement from queue i to queue j. Therefore we have in the SAN model a number of synchronizations which is equal to the number of non zero elements in the routing matrix P of the Jackson network we consider.

The local transitions are the external arrivals (rate λ_l) of customers, and the departures to the outside (rate μ_l multiplied by probability d_l). Assume that the master of synchronization r is l and the slave s. This synchronization describes a service in queue l followed by a transit from queue l to queue s. Therefore the rate of this synchronization is μ_l multiplied by probability $P(l, s)$. And we get:

$$\begin{array}{l} \mathbf{L_l} = \lambda_l(\mathbf{Upp} - \mathbf{I}) + \mu_l d_l(\mathbf{Low} - \mathbf{I^0}) \\ \mathbf{D^{(r)}} = \mu_l P(msr(r), sl(r))(\mathbf{Low} - \mathbf{I^0}), \\ \mathbf{E^{(r)}}_1 = \mathbf{Upp}. \end{array} \tag{8}$$

After substitution, we get for M_l:

$$M_l = (\lambda_l + \sum_{r=1}^{s} \Gamma_r 1_{sl(r)=l})(\mathbf{Upp} - \mathbf{I}) + \mu_l(d_l + \sum_{r=1}^{s} P(l, sl(r)) 1_{msr(r)=l})(\mathbf{Low} - \mathbf{I^0}).$$

Now, we take into account to simplify the expression of M_l that for all l we have: $d_l + \sum_{r=1}^{s} P(l, sl(r)) 1_{msr(r)=l} = 1$. Clearly for all l matrices M_l are tridiagonal. The equation $g_l M_l = 0$ can be solved very easily: g_l has a geometric distribution with rate ρ_l such that:

$$\rho_l = \frac{\lambda_l + \sum_{r=1}^{s} \Gamma^{(r)} 1_{sl(r)=l}}{\mu_l}.$$

Let us now consider the other equation of the fixed point system. If m is the master of synchronization r, then we must have: $\Gamma^{(r)}\ g_m = g_m \mu_m P(m, sl(r))$ **Low**. Therefore for all i (remember that the master of synchronization r is automaton m):

$$\Gamma_r g_m(i) = \mu_m P(m, sl(r)) g_m(i+1)$$

As g_m is geometric with ratio ρ_m, this relation becomes: $\Gamma_r = \rho_m \mu_m P(m, sl(r))$. After substitution we get:

$$\rho_l = \frac{\lambda_l + \sum_{r=1}^{s} \mu_m \rho_m P(m, sl(r)) 1_{sl(r)=l}}{\mu_l}.$$

This is the flow equation of a Jackson's network. Clearly g_l is also the marginal distribution found for this type of queueing network.

4.2 Gelenbe's Networks of Positive and Negative Customers

The concept of Generalized networks (G-networks for short) have been introduced by Gelenbe in [12]. These networks contain customers and signals. In the first papers on this topic, signals were also denoted as negative customers. Signals are not queued in the network. They are sent into a queue, and disappear instantaneously. But before they disappear they may act upon some customers present in the queue. As customers may, at the completion of their service, become signals and be routed into another queue, G-networks exhibit some synchronized transitions which are not modeled by Jackson networks. The first signal considered in [12] was described as a negative customer. A negative customer deletes an usual customer if there is any. These networks have a steady-state product form solution under usual Markovian assumptions. We consider an infinite state space. Each automaton models the number of positive customers in a queue. The signal are not represented in the states as they vanish instantaneously. The local transitions are the external arrivals (rate λ_l^+) of customers, the arrivals of negative customers with rates λ_l^- and the departures to the outside (rate μ_l multiplied by probability d_l). We have two types of synchronizations: The first type of synchronization describes the departure of a customer on the master (the end of service with rate μ_l and probability $P^-(msr(r), sl(r))$ and the arrival of a negative customers at the slave. The other type of synchronizations is the one used in Jackson networks: departure of a customer from the master and arrival as an usual customer at the slave. The rate is μ_l and probability $P^+(msr(r), sl(r))$. More formally:

$$\begin{array}{l} \mathbf{L_l} = \lambda_l^+(\mathbf{Upp} - \mathbf{I}) + \mu_l d_l(\mathbf{Low} - \mathbf{I^0}) + \lambda_l^-(\mathbf{Low} - \mathbf{I^0}), \\ \mathbf{D^{(r)}}_1 = \mu_l P^-(msr(r), sl(r)))(\mathbf{Low} - \mathbf{I^0}), \\ \mathbf{D^{(r)}}_2 = \mu_l P^+(msr(r), sl(r)))(\mathbf{Low} - \mathbf{I^0}), \\ \mathbf{E^{(r)}}_1 = \mathbf{Low} \quad and \quad \mathbf{E^{(r)}}_2 = \mathbf{Upp}. \end{array} \tag{9}$$

After substitution in the system considered in theorem 1, it must be clear that matrix M_l is again tridiagonal with constant diagonals. Thus, again g_l has a geometric distribution with rate ρ_l:

$$\rho_l = \frac{\lambda_l^+ + \sum_{r=1}^{s} \Gamma_2^{(r)} 1_{sl(r)=l}}{\mu_l + \lambda_l^- + \sum_{r=1}^{s} \Gamma_1^{(r)} 1_{sl(r)=l}}.$$

Of course, one must check that for all l, ρ_l is smaller than 1. Because of its geometric distribution, g_l is an eigenvector of operators $\overline{\mathbf{D^{(r)}}}_1$ and $\overline{\mathbf{D^{(r)}}}_2$. Finally, we obtain:

$$\Gamma_1^{(r)} = \rho_{msr(r)} \mu_{msr(r)} P^-(msr(r), sl(r)),$$

and

$$\Gamma_2^{(r)} = \rho_{msr(r)} \mu_{msr(r)} P^+(msr(r), sl(r)).$$

After substitution, we get the generalized flow equation which has been found in [12].

4.3 Gelenbe's Networks of Queues with Catastrophes

A catastrophe is a signal which flushes the customers out of the queue. Thus we have $\mathbf{E^{(r)}} = \mathbf{J0}$. The catastrophe has been studied in [7] with a multiclass model and can be defined using a model of batch deletion [13] where it can be assumed that the size of the batch of negative customers. The model is described by (for the sake of simplicity we assume that there is no movement of usual customers between the queues and at service completion the customers move and become signals):

$$\begin{array}{l} \mathbf{L_l} = \lambda_l^+(\mathbf{Upp} - \mathbf{I}) + \mu_l d_l(\mathbf{Low} - \mathbf{I^0}) \\ \mathbf{D^{(r)}} = \mu_l P^(msr(r), sl(r)))(\mathbf{Low} - \mathbf{I^0}), \\ \mathbf{E^{(r)}} = \mathbf{J0}. \end{array} \tag{10}$$

Matrix Fl has a tridiagonal structure plus the first column. The following property addresses the type of solution for such a matrix (the proof is omitted for the sake of concision).

Property 2. For all positive a, b and c, CTMC with transition rate matrix $(a(\mathbf{Upp}-\mathbf{I})+b(\mathbf{Low}-\mathbf{I^0})+c(\mathbf{J0}-\mathbf{I}))$ has a geometric steady-state distribution. The rate ρ of the geometric is the only one solution of $bX^2-(a+b+c)X+a=0$ which is between 0 and 1. Clearly such a solution already exists.

Therefore the marginal distribution of matrix F_l is geometric with rate ρ_l and the other equation implies that $\Gamma_r = \rho_l$. Again the fixed point system we found after substitution is equivalent to the one studied in [7].

4.4 Fixed Point System and Ergodicity

In the previous section we have mentioned that the existence of a fixed point system does not imply ergodicity of the CTMC associated to the SAN. Let us present now an example:

$$\begin{array}{l} \mathbf{L_l} = \mathbf{J0} - \mathbf{I} \\ \mathbf{D^{(r)}} = b(\mathbf{Low} - \mathbf{I^0}), \\ \mathbf{E^{(r)}} = a(\mathbf{Upp} - \mathbf{I}) + (1-a)(\mathbf{Low} - \mathbf{I^0}). \end{array} \tag{11}$$

Then M_l has the same structure we have mentioned before for a network with catastrophes. Therefore g_l has the same steady-state distribution. As $\mathbf{D^{(r)}}$ does not change, g_l is still an eigenvector and if the solution exists for a network of queues with catastrophes it also exists for the new system in Eq. 11. But if we consider the Markov chain associated with the SAN, it can easily be observed that it is not irreducible. Indeed, state $(0, 0, \ldots, 0)$ is now absorbing:

- local transitions are always possible but the automaton jumps to 0.
- synchronized transitions are not allowed when we are in state $(0, 0, \ldots, 0)$.

Thus the chain associated to the SAN is not ergodic.

4.5 Networks with Resets

In [14], the authors have presented a new type of signals denoted as reset which acts when the queue is empty. A reset makes the queue jumps to any state except 0 with a distribution which is closely related to the steady state distribution. If the queue is not empty at the arrival of a reset, it has no effect. Without loss of generality we assume there are no arrivals of resets from the outsides. The customers arrive from the outside with rate λ_l in queue l and as usual the service rate is μ_l, the departure probability d_l and we have two routing matrices P^+ for the routing of customers and P^- for the customers which join another queue as reset. The first synchronization is the transformation of a customer into a reset while the second one is the movement of customer. The network with customers and resets is defined by:

$$\begin{aligned}
&\mathbf{L_l} = \lambda_l(\mathbf{Upp} - \mathbf{I}) + \mu_l d_l(\mathbf{Low} - \mathbf{I^0}) + \Gamma_1^{(r)}(\mathbf{E^{(r)}}_1 - \mathbf{I}) + \Gamma_2^{(r)}(\mathbf{Upp} - \mathbf{I}),\\
&\mathbf{D^{(r)}}_1 = \mu_l P^-(msr(r), sl(r)))(\mathbf{Low} - \mathbf{I^0}),\\
&\mathbf{D^{(r)}}_2 = \mu_l P^+(msr(r), sl(r)))(\mathbf{Low} - \mathbf{I^0}),\\
&\mathbf{E^{(r)}}_2 = \mathbf{Upp}.
\end{aligned}$$

$$\mathbf{E^{(r)}}_1 = \begin{pmatrix}
0 & g_l(0) & g_l(1) & \ldots & g_l(n) \\
0 & 1 & 0 & 0\ 0\ 0 & 0 \\
0 & 0 & 1 & & \\
0 & & & 1 & \\
0 & & & & 1 \\
0 & & & & & 1 \\
0 & & & & & & 1
\end{pmatrix} \tag{12}$$

But due to the definition of $\mathbf{E^{(r)}}_1$ we have the following property:

Property 3. If distribution g_l is geometric, we have: $g_l(\mathbf{Upp} - \mathbf{I}) = g_l(\mathbf{E^{(r)}}_1 - \mathbf{I})$.

Proof: First perform the product of vector g_l by matrix $(\mathbf{Upp} - I)$:

$$g_l(\mathbf{Upp} - \mathbf{I}) = (-g_l(0), g_l(0) - g_l(1), g_l(1) - g_l(2), \ldots, g_l(n) - g_l(n+1), \ldots)$$

Remember that distribution g_l is geometric with ratio ρ_l. Therefore:

$$\begin{cases} g_l(0) - g_l(1) & = (1-\rho_l)^2 = g_l(0)^2 \\ g_l(1) - g_l(2) & = g_l(0)g_l(1) \\ g_l(2) - g_l(3) & = g_l(0)g_l(2) \\ \dots & \\ g_l(n) - g_l(n+1) & = g_l(0)g_l(n) \end{cases}$$

Therefore after substitution and factorization we get:

$$g_l(\mathbf{Upp} - \mathbf{I}) = g_l(0)(-1, g_l(0), g_l(1), \dots, g_l(n), \dots)$$

Now perform the product of vector g_l by matrix $(\mathbf{E^{(r)}}_1 - \mathbf{I})$:

$$g_l(\mathbf{E^{(r)}}_1 - \mathbf{I}) = g_l(0)(-1, g_l(0), g_l(1), \dots, g_l(n), \dots)$$

Both terms are equal. The equation is satisfied.

Let us now prove that this property implies the product form for that network with resets. We can rewrite the definition of $\mathbf{L_l}$ to substitute $g_l(\mathbf{E^{(r)}}_1 - \mathbf{I})$ by $g_l(\mathbf{Upp} - \mathbf{I})$. The new version of this equation is the same as the equation we found for a Jackson network.

$$\begin{cases} g_l \left(\lambda_l(\mathbf{Upp} - \mathbf{I}) + \mu_l(\mathbf{Low} - \mathbf{I^0}) + \Gamma_1^{(r)}(\mathbf{Upp} - \mathbf{I}) + \Gamma_2^{(r)}(\mathbf{Upp} - \mathbf{I}) \right) = 0 \\ \Gamma_1^{(r)} g_l = \mu_l P^-(msr(r), sl(r)))g_l(\mathbf{Low} - \mathbf{I^0}), \\ \Gamma_2^{(r)} g_l = \mu_l P^+(msr(r), sl(r)))g_l(\mathbf{Low} - \mathbf{I^0}), \end{cases} \tag{13}$$

These equations implies that the product form exist (see section 4.1) and the flow equation is similar to the one we get for a Jackson network.

4.6 First Generalization of Resets

Let us now generalize the former approaches. Assume that the model satisfy Theorem 1 and assume that g_l is geometric with rate ρ_l. Furthermore assume that $\mathbf{D^{(r)}} = a_r(\mathbf{Low} - \mathbf{I^0})$. After substitution we get:

$$\begin{cases} \Gamma^{(r)} = a_r \rho_l \quad if \quad msr(r) = l, \\ g_l \left[\mathbf{L_l} + \sum_{r=1}^{s} (\mathbf{Low} - \mathbf{I^0}) 1_{msr(r)=l} + \sum_{r=1}^{s} \Gamma^{(r)} 1_{sl(r)=l} (\mathbf{E^{(r)}} - \mathbf{I}) \right] = 0. \end{cases} \tag{14}$$

Now we add a new synchronization in that model. This synchronization is described by the same matrix we have already described for resets:

$$\mathbf{E^{(0)}} = \begin{pmatrix} 0 & g_l(0) & g_l(1) & \dots & g_l(n) \\ 0 & 1 & 0 & 0\ 0\ 0 & 0 \\ 0 & 0 & 1 & & \\ 0 & & & 1 & \\ 0 & & & & 1 \\ 0 & & & & & 1 \\ 0 & & & & & & 1 \end{pmatrix}.$$

Consider now the equation of the marginal and add the new synchronization.

$$g_l \left[\mathbf{L_l} + \sum_{r=1}^{s} a_r (\mathbf{Low} - \mathbf{I}) 1_{msr(r)=l} + \sum_{r=1}^{s} \Gamma^{(r)} (\mathbf{E^{(r)}} - \mathbf{I}) 1_{sl(r)=l} + \Gamma^{(0)} (\mathbf{E^{(0)}} - \mathbf{I}) 1_{sl(0)=l} \right]$$

But property 3 holds and we can substitute Ez by **Upp** in the equation on the marginal distribution of probability.

$$g_l \left[\mathbf{L_l} + \sum_{r=1}^{s} a_r (\mathbf{Low} - \mathbf{I}) 1_{msr(r)=l} + \sum_{r=1}^{s} \Gamma^{(r)} (\mathbf{E^{(r)}} - \mathbf{I}) 1_{sl(r)=l} + \Gamma^{(0)} (\mathbf{Upp} - \mathbf{I}) 1_{sl(0)=l} \right]$$

And this system is much easier to analyze than the former one. For instance, we obtain very easily that:

Property 4. Queueing networks with positive and negative customers and catastrophes and resets defined by matrix $\mathbf{E^{(0)}}$ have product form steady-state distribution if the rates of the marginal distributions (which are geometric) are smaller than one.

The proof consists only to verify that the system without resets has a geometric distribution and that the system with matrix $(\mathbf{Upp} - \mathbf{I})$ added in the marginal matrix still has a geometric vector in its kernel.

4.7 Second Generalization of Resets

Assume now that we have found a solution to the following fixed point equations:

$$\begin{cases} \Gamma^{(r)}\ g_l = g_l\ \overline{\mathbf{D^{(r)}}} \quad if \quad msr(r) = l, \\ g_l\ \left[\mathbf{L_l} + \sum_{r=1}^{s} \mathbf{D^{(r)}} 1_{msr(r)=l} + \sum_{r=1}^{s} \Gamma^{(r)} 1_{sl(r)=l} (\mathbf{E^{(r)}} - \mathbf{I}) \right] = 0. \end{cases} \tag{15}$$

Now we add some effects for a new signal. Let $\mathbf{E^{(0)}}$ the matrix description of this new signal. Assume that $\mathbf{E^{(0)}}$ satisfies:

$$\mathbf{E^{(0)}} = \left[\mathbf{I} + \frac{(\mathbf{L_l} + \sum_{r=1}^{s} \mathbf{D^{(r)}} 1_{msr(r)=l} + \sum_{r=1}^{s} \Gamma^{(r)} 1_{sl(r)=l} (\mathbf{E^{(r)}} - \mathbf{I}))}{\delta + \epsilon} \right]^k \tag{16}$$

where k is any positive integer, ϵ is positive and δ is the uniformization factor for matrix $(\mathbf{L_l} + \sum_{r=1}^{s} \mathbf{D^{(r)}} 1_{msr(r)=l})$. The summation on r does not take into account the new signal we add. Remember that (ϵ and k are dependent on r):

$$\delta = max_i(-\mathbf{L_l}[i,i] - \sum_{r=1}^{s} \mathbf{D^{(r)}}[i,i] 1_{msr(r)=l}).$$

Property 5. Assume that the system is based on equation 15 and add some signal with effect described by equation 16, then the fixed point system has the same solution.

Proof: Consider a solution of system 15. Clearly, this allows to simplify the first equation of the marginal:

$$g_l \Big[\mathbf{L_1} + \sum_{r=1}^{s} \mathbf{D^{(r)}} 1_{msr(r)=l} + \sum_{r=1}^{s} \Gamma^{(r)}(\mathbf{E^{(r)}} - \mathbf{I}) 1_{sl(r)=l} + \Gamma^{(0)}(\mathbf{E^{(0)}} - \mathbf{I}) 1_{sl(0)=l}\Big] = g_l \Gamma^{(0)}(\mathbf{E^{(0)}} - \mathbf{I}) 1_{sl(0)=l}.$$

Now, consider the definition of $\mathbf{E^{(0)}}$. We have:

$$g_l \left[\mathbf{I} + \frac{(\mathbf{L_1} + \sum_{r=1}^{s} \mathbf{D^{(r)}} 1_{msr(r)=l} + \sum_{r=1}^{s} \Gamma^{(r)}(\mathbf{E^{(r)}} - \mathbf{I}) 1_{sl(r)=l})}{\delta + \epsilon}\right] = g_l.$$

Therefore, $g_l \mathbf{E^{(0)}} = 0$. Finally:

$$g_l \left[\mathbf{L_1} + \sum_{r=1}^{s} \mathbf{D^{(r)}} 1_{msr(r)=l} + \sum_{r=1}^{s} \Gamma^{(r)}(\mathbf{E^{(r)}} - \mathbf{I}) 1_{sl(r)=l} + \Gamma^{(0)}(\mathbf{E^{(0)}} - \mathbf{I}) 1_{sl(0)=l}\right] = 0,$$

and $\Gamma^{(r)}$ is still the same. Therefore resets defined by this family of matrix $\mathbf{E^{(0)}}$ preserve the steady-state product distribution.

5 Conclusions

The theorem we prove here allows to generalize many results on queues with customers and signals. But the post important result is a simple algebraic proof based on tensor. As the tensor representation is not limited to SAN (see for instance [5,20]). We hope that this approach will lead to new research activities on the link between tensor representation and closed form solutions.

Acknowledgement. This work was partially supported by ANR grant SMS (ANR-05-BLAN-0009-02) and ANR SETIN Checkbound.

References

1. Boucherie, R.: A Characterization of independence for competing Markov chains with applications to stochastic Petri nets. IEEE Trans. Software Eng. 20(7), 536–544 (1994)
2. Buchholz, P., Dayar, T.: Comparison of Multilevel Methods for Kronecker-based Markovian Representations. Computing Journal 73(4), 349–371 (2004)
3. Dayar, T., Gusak, O., Fourneau, J.M.: Stochastic Automata Networks and Near Complete Decomposability. SIAM Journal on Matrix Analysis and Applications 23, 581–599 (2002)
4. Dao-Thi, T.-H., Mairesse, J.: Zero-Automatic Networks. Discrete Event Dynamic Systems 18(4), 499–536 (2008)
5. Donnatelli, S.: Superposed stochastic automata: a class of stochastic Petri nets with parallel solution and distributed state space. Performance Evaluation 18, 21–36 (1993)

6. Fernandes, P., Plateau, B., Stewart, W.J.: Efficient Descriptor-Vector Multiplications in Stochastic Automata Networks. In: JACM, pp. 381–414 (1998)
7. Fourneau, J.M., Kloul, L., Quessette, F.: Multiple class G-networks with jumps back to Zero. IEEE Mascots 95, 28–32 (1995)
8. Fourneau, J.M., Plateau, B., Stewart, W.: Product form for Stochastic Automata Networks. In: Proc. of ValueTools 2007, Nantes, France (2007)
9. Fourneau, J.M., Plateau, B., Stewart, W.: An Algebraic Condition for Product Form in Stochastic Automata Networks without Synchronizations. Performance Evaluation 65(11-12), 854–868 (2008)
10. Fourneau, J.M.: Discrete Time Markov chains competing over resources: product form steady-state distribution. In: IEEE QEST 2008, pp. 147-156 (2008)
11. Fourneau, J.-M.: Product Form Steady-State Distribution for Stochastic Automata Networks with Domino Synchronizations. In: Thomas, N., Juiz, C. (eds.) EPEW 2008. LNCS, vol. 5261, pp. 110–124. Springer, Heidelberg (2008)
12. Gelenbe, E.: Product form queueing networks with negative and positive customers. Journal of Applied Probability 28, 656–663 (1991)
13. Gelenbe, E.: G-Networks with signals and batch removal. Probability in the Engineering and Informational Sciences 7, 335–342 (1993)
14. Gelenbe, E., Fourneau, J.M.: G-networks with resets. Performance Evaluation 49(1/4), 179–191 (2002)
15. Harrison, P.G., Hillston, J.: Exploiting Quasi-reversible Structures in Markovian Process Algebra Models. Computer Journal 38(7), 510–520 (1995)
16. Harrison, P.G.: Turning back time in Markovian process algebra. Theoretical Computer Science 290, 1947–1986 (2003)
17. Harrison, P.G.: Reversed processes, product forms and a non-product form. Journal of Linear Algebra and Applications 386, 359–381 (2004)
18. Harrison, P.G.: Compositional reversed markov processes with applications to G-networks. Performance Evaluation 57, 379–408 (2004)
19. Hillston, J., Thomas, N.: Product Form Solution for a Class of PEPA Models. Performance Evaluation 35(3-4), 171–192 (1999)
20. Kloul, L., Hillston, J.: An efficient Kronecker representation for PEPA models. In: de Luca, L., Gilmore, S. (eds.) PROBMIV 2001, PAPM-PROBMIV 2001, and PAPM 2001. LNCS, vol. 2165, p. 120. Springer, Heidelberg (2001)
21. Plateau, B.: On the Stochastic Structure of Parallelism and Synchronization Models for Distributed Algorithms. In: Proc. ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, Austin, Texas (August 1985)
22. Plateau, B., Fourneau, J.M., Lee, K.H.: PEPS: A Package for Solving Complex Markov Models of Parallel Systems. In: Proceedings of the 4th Int. Conf. on Modeling Techniques and Tools for Computer Performance Evaluation, Majorca (1988)
23. Stewart, W.J., Atif, K., Plateau, B.: The numerical solution of Stochastic Automata Networks. European Journal of Operation Research 86(3), 503–525 (1995)

Weak Stochastic Comparisons for Performability Verification*

Hind Castel-Taleb[1] and Nihal Pekergin[2]

[1] INSTITUT TELECOM, Telecom et Management SudParis
9, rue Charles Fourier 91011 Evry Cedex, France
hind.castel@int-evry.fr
[2] LACL, Université Paris-Est Val de Marne
61, av. du Général de Gaulle, 94010 Créteil Cedex, France
nihal.pekergin@univ-paris12.fr

Abstract. The probabilistic model checking provides a precise formalism for the performance and reliability verification of telecommunication systems modeled by Markov chains. We study a queueing system similar to a Jackson network except that queues have a finite capacity. We propose to study in this paper (state and path) formulas from the Continuous Stochastic Logic (CSL), in order to verify performability properties. Unfortunately, transient and stationary analysis is very complex for multidimensional Markov processes. So we propose to use the stochastic comparisons in the sense of weak orderings to define bounding processes. Bounding processes are represented by independent M/M/1 queues for which transient and stationary distributions can be computed as the product of probability distributions of each queue. We use the increasing set method, and we develop an intuitive formalism based on events to establish weak stochastic comparisons.

1 Introduction

Probabilistic model checking is an efficient method for the verification of performance and reliability properties on telecommunication and computer networks systems. CSL (Continuous Stochastic Logic) [1,2] formalism allows us to check transient and stationary properties of the considered system modelled by a CTMC (Continuous Time Markov Chain). CSL formulas are computed from transient or the steady-state probability distributions of the underlying chain. If there is not a specific form for (the stationary and transient) probability distributions then it could be difficult to compute them for multidimensional cases due to the state space explosion. For particular processes as independent M/M/1 queues, both stationary and transient probability distributions can be derived easily. We propose in this paper to study a queueing system similar to a Jackson network except that queues have a finite capacity, in order to verify some property occurrences such as congestions, availability, etc.. This queueing system does

* Partially supported by french research project ANR-SETI06-02.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 294–308, 2009.

not have product-form solution thus it is difficult to analyze. Thus we propose to bound it by a process defined as independent M/M/1 queues which is easier to study. Establishing stochastic bounds in the case of multidimensional state spaces is complex. Several stochastic orderings can be defined corresponding to different comparison relations of the underlying distributions. There are different methods to compare processes: increasing sets, and the coupling. Increasing set methods [10], [11] is a general formalism, allowing the definition of the strong stochastic ordering $\preceq_{st}$, and weaker orderings as $\preceq_{wk}$, and $\preceq_{wk^*}$. The coupling method [8] is a more direct method, but it allows to define only the $\preceq_{st}$ ordering. The two methods lead to the comparison of transition rates but the approach used is different: one with the coupling of sample paths, and the other with the definition of families of increasing sets.

Bounding methods are suitable in model checking, since we need to check if some constraints are satisfied or not without considering exact values. The stochastic comparison approach provides an interesting alternative for model checking since this approach lets us to provide the bounds on transient distributions as well as the stationary distribution of the underlying Markovian model. Indeed, the stochastic comparison of distributions provides the inequalities on the partial sum of probabilities. In model checking, given a formula, the verification is resumed to compute the sum of probabilities of states satisfying this formula in a transient or the stationary distribution. The verification through bounding models depends on the comparison operator, let explain it for the case to check if an upper bound is satisfied ($\leq p$). Instead of computing the sum of probabilities over states satisfying the considered property (formula), we compute bounds B_{inf} and B_{sup} considering bounding models. There are 3 possible decisions: i. If $B^{sup} \leq p$ then we can decide that formula is satisfied. ii. If $B^{inf} > p$ then we can decide that formula is not satisfied. iii. otherwise it is not possible to decide with these bounding values, the bounding models must be refined if it is possible.

Related works

There are some interesting studies about the model checking of multidimensional CTMCs. In [13], CSL model checking on infinite state spaces in the case of Jackson queueing networks is studied. A general approach based on finite CTMCs and QBDs is adopted for CSL path formulas. For the case of finite and large state spaces, the lumping equivalence is used in order to reduce Markov chains state spaces [2]. The notion of state equivalence relation also called stochastic bisimulation is introduced in order to build a reduced state space called *quotient*. The relation is based on the required conditions that must have equivalent states as equality of state labeling, rewards, exit rates. The advantage is the verification of CSL logic formulas on a reduced state space. Stochastic comparison method has been also used for the model checking of complex system. In [12], it has been proposed to check state formulas defined over Discrete Time Markov Chain (DTMC) rewards. The authors have assumed a total order on the state space, and generate the aggregated Markov chains using the LIMSUB algorithm [7], based on lumpability constraints. In [4], we suppose a partial order which

allows tighter comparison conditions for the definition of aggregated bounding Markov processes. A parametric aggregation scheme is proposed in order to improve the quality of the bounds so the precision of the checking procedure. In [3], the model checking approach for CSL logic is studied using class C bounding Markov chains, which have closed-form solutions for transient and steady state distribution. For more details, [15] presents in details the stochastic comparison methods for the model checking of complex Markov chains. Although stochastic bounding method provides an efficient technique in the model checking context, most studies are based on the strong stochastic ordering which implies severe constraints between the compared processes. We propose in this paper to use the weak ordering in order to improve the quality of the bounds. Thus we increase the number of cases for which we can conclude by this approach. This paper has two main objectives: first, we develop the stochastic comparison based on increasing sets and we provide an intuitive approach using events. Secondly we apply the model checking on bounding Markov processes by evaluating states and paths formulas. We present an application on a telecommunication system modelled as a queueing network in order to verify performance properties. This system is equivalent to a Jackson network except that queues have a finite capacity. Unfortunately, quantitative analysis of this system for stationary or steady-state is very difficult. We propose to bound it by a process represented by independent M/M/1 queues which is easier to analyze. We propose to compute steady state and path formulas from the CSL logic on bounding systems instead of the original one in order to perform the verification.

This paper is organized as follows: first, we present basic notions of the stochastic ordering theory, precisely increasing sets method for weak stochastic comparisons. Section 3 introduces the systems to compare, and presents the increasing sets formalism for the stochastic comparisons of processes (CTMC). Note that the monotonicity is also presented in order to prove the "weak" monotonicity of independent M/M/1 queues. Section 4 is devoted to the CSL model checking, we explain how to apply weak stochastic comparisons for the verification of steady-state and path formulas.

2 Stochastic Comparison Method

Quantitative analysis of multidimensional Markov processes could be very difficult for stationary and transient analysis. In order to solve this problem, we propose to bound the original Markov process by another Markov process which is easier to analyze, in order to compute performance measure bounds. Stochastic comparison method is based on the stochastic ordering theory, which is presented just after.

2.1 Stochastic Ordering Theory

Let E be a discrete, and countable state space, and $\preceq$ be at least a preorder (reflexive, transitive but not necessarily an anti-symmetric binary relation) on

E. We suppose that E is a multidimensional state space, where each component is discrete, as it is generally the case in the queueing models. Several stochastic orderings related to increasing sets (functions) can be defined when the state space is partially ordered. The most known is the strong stochastic ordering $\preceq_{st}$, but also weaker orderings can be defined: $\preceq_{wk}$, and $\preceq_{wk^*}$ [10]. The strong stochastic ordering is equivalent to a sample path ordering, the $\preceq_{wk}$ ordering to a tail distributions comparison, and $\preceq_{wk^*}$ serve the same role for cumulative distribution functions. Different formalisms can be used to define a stochastic ordering: increasing functions, and increasing sets [14]. We focus on the increasing set formalism in this paper, since it will be used to establish the comparison of processes. Let us remark here that we apply the term process to mean continuous time Markov chains in the sequel.

2.2 Increasing Set Formalism

First, we define an increasing set. Let $\Gamma \subseteq E$, we denote by

$$\Gamma \uparrow = \{y \in E \mid y \succeq x, x \in \Gamma\} \tag{1}$$

Definition 1. *Γ is called an increasing set if and only if $\Gamma = \Gamma \uparrow$*

From the general definition of an increasing set, [10] has defined the family $\Phi(E)$ of increasing sets generating the stochastic ordering $\preceq_{\Phi}$. Three families of increasing sets are defined. The first one is $\Phi_{st}(E)$ which is defined from all the increasing sets of E:

$$\Phi_{st}(E) = \{\text{all increasing sets on } E\} \tag{2}$$

$\Phi_{st}(E)$ induces the $\preceq_{st}$ ordering. In the same way, the stochastic orderings $\preceq_{wk}$ and $\preceq_{wk^*}$ are defined respectively from the families $\Phi_{wk}(E)$ and $\Phi_{wk^*}(E)$ by taking particular kinds of increasing sets [10].

$$\Phi_{wk}(E) = \{\{x\} \uparrow,\ x \in E\} \tag{3}$$

and

$$\Phi_{wk^*}(E) = \{E - \{x\} \downarrow,\ x \in E\} \tag{4}$$

Let X and Y be two random variables defined on E, and their probability measures given respectively by the probability vectors p and q where $p[i] = Prob(X = i),\ \forall i \in E$ (resp. $q[i] = Prob(Y = i),\ \forall i \in E$).

Definition 2

$$X \preceq_{\Phi} Y \Leftrightarrow \sum_{x \in \Gamma} p[x] \leq \sum_{x \in \Gamma} q[x],\ \forall \Gamma \in \Phi(E) \tag{5}$$

Next, we present the stochastic comparison of Markov processes.

Let $\{X(t), t \geq 0\}$ (resp. $\{Y(t), t \geq 0\}$) a CTMC defined on E. We will compare stochastically $X(t)$ with $Y(t)$ using a stochastic ordering $\preceq_{\Phi}$ ($\preceq_{st}$, $\preceq_{wk}$, $\preceq_{wk^*}$) [14], [10].

Definition 3. *We say that*

$$\{X(t), t \geq 0\} \preceq_{\Phi} \{Y(t), t \geq 0\} \tag{6}$$

$$if X(0) \preceq_{\Phi} Y(0) \Longrightarrow X(t) \preceq_{\Phi} Y(t), \forall t > 0 \tag{7}$$

If we suppose that $\{X(t),\ t \geq 0\}$ (resp. $\{Y(t),\ t \geq 0\}$) is a CTMC with infinitesimal generator matrix Q_1 (resp. Q_2), then we present the theorem of the stochastic comparison of CTMC using increasing set formalism [10], [14].

Theorem 1. *We say that:*

$$\{X(t), t \geq 0\} \preceq_{\Phi} \{Y(t), t \geq 0\} \tag{8}$$

if and only if the following conditions are verified:

1. $X(0) \preceq_{\Phi} Y(0)$
2. $\{X(t), t \geq 0\}$ *or* $\{Y(t), t \geq 0\}$ *is* $\preceq_{\Phi}$*-monotone*
3. $$\forall x \in E,\ \sum_{z \in \Gamma} Q_1(x,z) \leq \sum_{z \in \Gamma} Q_2(x,z),\ \forall \Gamma \in \Phi(E) \tag{9}$$

The monotonicity is one of the sufficient conditions of this theorem and it means that, depending on the initial condition, the process is increasing or decreasing in time.

corresponds to an increasing in time of a process. Next we give the definition of the $\preceq_{\Phi}$-monotonicity.

Definition 4. *We say that* $\{X(t), t \geq 0\}$ *is* $\preceq_{\Phi}$ *−monotone if*

$$X(t) \preceq_{\Phi} (\succeq_{\Phi}) X(t+\tau), \forall t \geq 0, \forall \tau \geq 0 \tag{10}$$

The $\preceq_{st}$-monotonicity can be proved by the coupling of the process with itself [8,9], or from the increasing sets formalism generating transition rate comparisons [10]. But the coupling can be used only for the $\preceq_{st}$ ordering, and there is no result about transition rates comparisons for the $\preceq_{wk}$ ordering. Next, we try to generalize the transition rates comparison [10] to any order $\preceq_{\Phi}$ ($\preceq_{st}$, $\preceq_{wk}$, $\preceq_{wk^*}$).

Theorem 2. *If the following condition is verified:*

$$\forall \Gamma \in \Phi(E),\ \sum_{z \in \Gamma} Q_1(x,z) \leq \sum_{z \in \Gamma} Q_1(y,z),\ x \preceq y \in E,\ x,y \in \Gamma \text{ or } x,y \notin \Gamma \tag{11}$$

then process $X(t)$*is* $\preceq_{\Phi}$*-monotone.*

The proof of this theorem is given in [5].

3 Stochastic Comparison Application

In this section, we present the exact and the bounding systems. We give the proof of the stochastic comparison using increasing set formalism, for the $\preceq_{wk}$ ordering.

3.1 Considered Systems

The system understudy is similar to a Jackson network with n finite capacity queues. Let $\{X(t), t \geq 0\}$ be the (CTMC) representing the evolution of this system, and Q the infinitesimal generator. We denote by Π the stationnary probability distribution. Each queue i has a finite capacity B_i, and is characterized by the following parameters:

- Exponential inter-arrival times, with parameters λ_i
- Exponential service times, with parameters μ_i, and after the service, we have:
  - with the probability p_{ij} the customer transits from queue i to queue j if it is not full
  - with the probability d_i the customer goes out.

As queues have a finite capacity, then a customer arriving in a full queue is lost. $\{X(t), t \geq 0\}$ is a multidimensional CTMC and there is no product form solution to compute the steady-state nor transient distributions. We propose to use the stochastic comparisons to derive bounding models to consider CSL path and state formulas. This bounding model is defined on $E = \mathbb{N}^n$. We consider the widely used component-wise partial ordering $\preceq$ on this state space:

$$\forall x, y \in \mathbb{N}^n,\ x \preceq y \Leftrightarrow x_i \leq y_i, \forall i = 1, \ldots, n \tag{12}$$

We propose to bound $\{X(t), t \geq 0\}$ by the CTMC $\{X^u(t), t \geq 0\}$ defined by n independent M/M/1 queues. Independent queues are defined by deleting links between queues, and by adding transit flow to the arrivals in the queues. Each queue i has a finite capacity B_i, with the following parameters: arrival rates $\lambda_i + \sum_{j=1}^{n} \mu_j p_{ji}$, and service rate μ_i. This process is interesting as the stationary and transient probability distribution can be computed as the product of probability distributions of M/M/1 queues. We denote by Q^u the infinitesimal generator, and Π^u the stationnary probability distribution. In [10], [11], a similar study has been presented with infinite capacities queues. First, the $\preceq_{st}$ ordering is not possible because using the coupling of the sample path we can see that the rate to go outside $\mu_i d_i$ of the exact process is lower than the rate decrease μ_i of the M/M/1 queues. This means that in the upper bounding model the decrease rate is greater which is a contradiction. In [10,11], an operator approach has been used in order to prove that the weak ordering could be defined. In the present paper, we suppose that systems have a finite capacity, and we develop the increasing set formalism governed with events for the weak ordering $\preceq_{wk}$.

3.2 Stochastic Comparison with Weak Orderings

We will explain in this section how to compare $X(t)$ and $X^u(t)$ using the increasing set formalism. We apply theorem 1, for the $\preceq_{wk}$ ordering. The increasing set theory is not easy to apply because the stochastic comparison is performed on all the increasing sets of a family. For multidimensional state spaces, as the state space increases exponentially, then the number of increasing sets will be also very large. We propose to solve this problem by defining only the increasing sets which are necessary to the comparison.

In theorem 1, there are two steps: first we must verify the monotonicity of one of the processes (condition 2), and secondly, we have to compare the transition rates of the processes (condition 3). We begin with the monotonicity, and we apply theorem 2 in order to verify if the process represented by independent M/M/1 queues is $\preceq_{wk}$-monotone. We choose this process as it is easier to analyze (with less events).

"$\preceq_{wk}$" Monotonicity of the Independent M/M/1 Queues. As E is multidimensional, then the set $\Phi_{wk}(E)$ could be very large. So we need to define the increasing sets which are used for the $\preceq_{wk}$- monotonicity. And we have to prove that transitions rates from states x and y, to states in increasing sets are verified. As these transitions are triggered by events, we define the increasing sets from the states x, y and these events. From a state x, in a queue i, we can have an arrival, or a service or nothing.

Let e_i be a binary vector on $\{1, \ldots, n\}$, where all the components are null except the component i which equals 1. This vector will be used to represents the evolution of the process from a state x after an event. For example, with an arrival in queue i, we have a transition from state x to $x + e_i$. So the increasing sets used for the monotonicity are:

$$\{x\}\uparrow, \{x + e_i\}\uparrow, \{x - e_i\}\uparrow, \{y\}\uparrow, \{y + e_i\}\uparrow, \{y - e_i\}\uparrow \tag{13}$$

Since we must also take the condition:

$$x, y \in \Gamma \text{ or } x, y \notin \Gamma \tag{14}$$

we do not have to take the increasing set: $\{y\}\uparrow$ as x is not in this increasing set. We denote by $S_{wk}(E)$ the set of increasing states which are sufficient for the comparison. It is defined by:

$$S_{wk}(E) = \{\{x\}\uparrow, \{x + e_i\}\uparrow, \{y + e_i\}\uparrow, \{x - e_i\}\uparrow, \{y - e_i\}\uparrow\} \tag{15}$$

Next, we define each increasing set.

Increasing sets definition: We have three constraints to use: the condition $x \preceq y$, the events, and the condition $x, y \in \Gamma$ or $x, y \notin \Gamma$.

We give for each increasing set the list of states to which the transitions are not null. The three dots $(\ldots)$ in the sets means that there are others states, but we don't need to give them as transitions are null.

– For $\{x + e_i\}\uparrow$, we need to define states which are greater than $x + e_i$. In this case, if $x_i < y_i$ we have : $x + e_i$, y, $y + e_i$ (as $y \succeq x + e_i$, and $y + e_i \succeq x + e_i$). As condition (14) will not be verified, then we don't take this case. If $x_i = y_i$, then we have the states: $x + e_i$, $y + e_i$, and so the condition 14 will be verified. So if $x_i < B_i$ and $y_i < B_i$:

$$\{x + e_i\}\uparrow = \{x + e_i, \ldots, y + e_i, \ldots\} \tag{16}$$

– For $\{y + e_i\}\uparrow$, we need to define states which are greater than $y + e_i$, so we have only $y + e_i$. If $y_i < B_i$, then:

$$\{y + e_i\}\uparrow = \{y + e_i, ...\} \tag{17}$$

– For $\{x\}\uparrow$, we need to define states which are greater than x. So we have x, we have also y as $x \preceq y$, and we can have also some states $y - e_k$, $k = 1, \ldots, n$ such that $y - e_k \succeq x$, we have also $x + e_k (k = 1 \ldots n$, and $y + e_k (k = 1, \ldots n)$. So we have:

$$\{x\}\uparrow = \{x, \ldots y - e_k \ (k = 1, \ldots, n \ if \ y_k > 0 \ and \ y - e_k \succeq x) \ldots, y, \ldots,$$

$$x + e_k (k = 1 \ldots n, \ if \ x_k < B_k), \ldots, y + e_k (k = 1, \ \ldots n \ if \ y_k < B_k)\} \tag{18}$$

– For $\{x - e_i\}\uparrow$, we obtain if $x_i > 0$

$$\{x - e_i\}\uparrow = \{x - e_i, \ldots, y - e_k (k = 1 \ldots n, \ y - e_k \geq x - e_i), \ldots,$$

$$x, \ldots y, \ldots, x + e_k (k = 1 \ldots n, \ x_k < B_k), \ldots y + e_k (k = 1 \ldots n, \ y_k < B_k)\} \tag{19}$$

– For $\{y - e_i\}\uparrow$, we have y in the set, but we are not sure to have x. In order to have the condition (14) verified, then we take the case where x is also in the increasing set which could be true if $y - e_i = x$ (so in the set we write only one of them: $y - e_i$) . If $y_i > 0$, then

$$\{y - e_i\}\uparrow = \{y - e_i, \ldots, y, \ldots x + e_k (k = 1 \ldots n, \ x + e_k < B_k), \ldots,$$

$$y + e_k (k = 1 \ldots n, \ y_k < B_k)\} \ if \ y_i > 0 \tag{20}$$

We compute now the transition rates in these increasing sets. In order to simplify the notation, we denote as follows the increasing sets: $\Gamma_{x+e_i} = \{x + e_i\}\uparrow$, $\Gamma_x = \{x\}\uparrow$, $\Gamma_{x-e_i} = \{x - e_i\}\uparrow$, $\Gamma_{y+e_i} = \{y + e_i\}\uparrow$, $\Gamma_{y-e_i} = \{x - e_i\}\uparrow$.

In the case of frontier states, for example x (resp y) is such that $x_i = B_i$ (resp $y_i = B_i$), then only increasing sets $\Gamma_x = \{x\}\uparrow$, $\Gamma_{x-e_i} = \{x - e_i\}\uparrow$, $\Gamma_{y-e_i} = \{y - e_i\}\uparrow$ are necessary for the comparison. In the case of only y is such that $y_i = B_i$, then we use increasing sets Γ_x, Γ_{x-e_i}, and Γ_{y-e_i}. In the case of $x_i = y_i = 0$, then we use $\Gamma_{x+e_i} = \{x + e_i\}\uparrow$, $\Gamma_{y+e_i} = \{y + e_i\}\uparrow$, $\Gamma_x = \{x\}\uparrow$. Now, as we have defined the increasing sets, we can compute the transition rates of the processes in order to compare them.

Transition rates comparison: For each increasing set, we compute the transition rates $\sum_{z\in\Gamma} Q^u(x,z)$ and $\sum_{z\in\Gamma} Q^u(y,z)$ for $\Gamma \in S_{wk}(E)$, in order to compare them.

Γ	$\sum_{z\in\Gamma} Q^u(x,z)$	$\sum_{z\in\Gamma} Q^u(y,z)$
Γ_{x+e_i}	$\lambda_i + \sum_{j\neq i}\mu_j p_{ji}$	$\lambda_i + \sum_{j\neq i}\mu_j p_{ji}$
Γ_{y+e_i}	0	$\lambda_i + \sum_{j\neq i}\mu_j p_{ji}$
Γ_x	$-\sum_{k=1}^n \mu_k 1_{x_k>0}$	$-\sum_{k=1}^n \mu_k 1_{y_k>0}1_{y_k=x_k}$
Γ_{x-e_i}	$-\sum_{k\neq i} \mu_k 1_{x_k>0}$	$-\sum_{k\neq i} \mu_k 1_{y_k>0}1_{y_k=x_k}$
Γ_{y-e_i}	$-\sum_{k} \mu_k 1_{x_k>0}$	$-\sum_{k\neq i} \mu_k 1_{y_k>0}$

For the case where $x = y$, then we choose only increasing sets defined from x or y, for example $\{x+e_i\}$, $\{x\}$, $\{x-e_i\}$, and in this case we can deduce easily that the transition rates are equal.

In the case where $x \prec y$, the comparison of the transition rates is easy for increasing sets Γ_{x+e_i} and Γ_{y+e_i}. For others increasing sets, we need to explain how to compare transition rates. Lets compare $-\sum_{k=1}^n \mu_k 1_{x_k>0}$ with $-\sum_{k=1}^n \mu_k 1_{y_k>0}1_{y_k=x_k}$. We need to compare the term:

$$\mu_k 1_{x_k>0} \qquad \text{with} \qquad \mu_k 1_{y_k>0}1_{y_k=x_k}$$

As : $1_{y_k>0}1_{y_k=x_k} = 1_{x_k>0}1_{y_k=x_k}$ and : $1_{x_k>0}1_{y_k=x_k} \leq 1_{x_k>0}$,
then we have the comparison of the sum:

$$-\sum_{k=1}^n \mu_k 1_{x_k>0} \leq -\sum_{k=1}^n \mu_k 1_{y_k>0}1_{y_k=x_k} \tag{21}$$

For increasing sets Γ_{x-e_i} and Γ_{y-e_i} the comparison is similar. So we can deduce that:

$$\forall x \preceq y \mid x,y \in \Gamma, \text{ or } x,y \notin \Gamma \tag{22}$$

$$\forall \Gamma \in S_{wk}(E), \sum_{z\in\Gamma} Q^u(x,z) \leq \sum_{z\in\Gamma} Q^u(y,z) \tag{23}$$

We conclude that $X^u(t)$ is $\preceq_{wk}$-monotone. We can apply theorem 1, and precisely the comparison of the transition rates of each process in order to perform the comparison.

Generator Comparison with Weak Orderings. We give briefly the proof of the $\preceq_{wk}$-comparison of $X(t)$ and $X^u(t)$. We apply theorem 1: so we compare $\sum_{z\in\Gamma} Q(x,z)$ and $\sum_{z\in\Gamma} Q^u(x,z)$ for increasing sets Γ_{x+e_i}, $\Gamma_{x-e_j+e_i}$, Γ_x, Γ_{x-e_i}.

Γ	$\sum_{z\in\Gamma} Q(x,z)$	$\sum_{z\in\Gamma} Q^u(x,z)$
Γ_{x+e_i}	λ_i	$\lambda_i + \sum_{k\neq i}\mu_k p_{ki}$
$\Gamma_{x-e_j+e_i}$	$\mu_j p_{ji} + \lambda_i$	$\lambda_i + \sum_{k\neq i}\mu_k p_{ki}$
Γ_x	$-\sum_{k=1}^n \mu_k 1_{x_k>0}$	$-\sum_{k=1}^n \mu_k 1_{x_k>0}$
Γ_{x-e_i}	$-\sum_{k\neq i} \mu_k 1_{x_k>0}$	$-\sum_{k\neq i} \mu_k 1_{x_k>0}$

So we can deduce from theorem 1 that: $\{X(t), t \geq 0\} \preceq_{wk} \{X^u(t), t \geq 0\}$ which means that:

$$P(X(t) \in \Gamma) \leq P(X^u(t) \in \Gamma), \forall \Gamma \in \Phi_{wk}(E) \tag{24}$$

It is easy to see that $\forall \Gamma \in \Phi_{wk}(E)$, as $\Gamma = \{x\}\uparrow$, and $x = (x_1, \ldots, x_n)$, then $\Gamma = \times_{i=1}^{n} \Gamma_i$, where $\Gamma_i = \{x_i\}\uparrow$. We deduce that $\forall x = (x_1, \ldots, x_n)$:

$$P(X(t) \succeq x) \leq \Pi_{i=1}^{n} P(X_i^u(t) \succeq x_i) \tag{25}$$

Where $X_i^u(t)$ is the Markov process representing evolution of the M/M/1 queue i. So we can bound the transient behavior of $X(t)$ by the product of the transient behavior of M/M/1 queues. Note that as we have defined an upper bound for $X(t)$, we can also define a lower bound $X^l(t)$ with n independent queues, where each queue i is represented by an arrival rate λ_i, and a service rate μ_i. We can prove easily the monotonicity of this process, and also the comparison of the generators. Next we explain how to apply equation 25 to verify some path formulas of CSL model checking.

4 CSL Model Checking Using Stochastic Comparisons

Continuous Stochastic Logic (CSL) is an extension of Computation Tree Logic (CTL) with two probabilistic operators that refer to the steady-state and transient behaviors of the underlying system. It is a branching-time temporal logic with state and path formulas defined for the considered model which is a continuous time Markov chain (CTMC) [1,2]. The states formulas are interpreted over states of the CTMC, while path formulas are interpreted over paths of the CTMC. We assume that each state is labelled with atomic propositions. Atomic propositions identify specific situations of the system such as "buffer full", "buffer empty" or "variable X is positive". Let I be an interval on the real line, p a probability and $\triangleleft$, a comparison operator ($\triangleleft \in \{<, \leq, >, \geq\}$). The syntax of CSL is defined by the following grammar [9]:

State-formulas:

$$\phi ::= true \mid a \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \mathcal{P}_{\triangleleft p}(\psi) \mid \mathcal{S}_{\triangleleft p}(\phi)$$

Path-formulas:

$$\psi ::= \phi_1 \, \mathcal{U}^I \phi_2 \mid \mathcal{X}^I(\phi)$$

For the state formulas $\neg$, $\vee$ and $\wedge$ the meanings are as usual in logic. In order to express the time span of a certain path, the path operators until ($\mathcal{U}$) and next ($\mathcal{X}$) use the time interval I [2]. The next operator $\mathcal{X}^I(\phi)$ states that a transition to a ϕ-state is made in the time interval I. The until operator $\phi_1 \, \mathcal{U}^I \phi_2$ asserts that ϕ_2 is satisfied in the time interval I and that all preceding time instants ϕ_1 holds.

The probabilistic operator $\mathcal{P}_{\triangleleft p}(\phi)$, is valid in state s (written as $s \models \mathcal{P}_{\triangleleft p}(\phi)$) if the probability measure of the set of paths starting in s and satisfying path

formula ϕ meets the bound $\lhd p$. The steady state operator $\mathcal{S}_{\lhd p}(\phi)$ denotes that the steady state probability for ϕ-states meets the bound p. We explain in this section how we use the $\preceq_{wk}$ comparisons for CSL model checking. Let us first remark here that we consider the case of the upper bounding threshold $\leq p$ ($< p$), since we build upper bounding models. Thus we can conclude that a CSL property ϕ (for example $\mathcal{P}_{\leq p}(\psi)$ for a given initial state s_0) is satisfied if the sum of states satisfying ϕ (the sum of probabilities over paths initiated from s_0 passing through states ψ) is less than p. The lower bounds on the threshold p ($>$,$\geq$) can be also considered since the satisfaction of ϕ is $\leq p$ then the satisfaction of property $\neg\phi$ is $> 1 - p$.

4.1 Checking $\mathcal{S}_{\lhd p}(\phi)$

For this formula, let us to check if the property is satisfied in the long run. Thus one must compute the steady-state probability distribution and then sum the probabilities that satisfy this property. In the case where the underlying model is ergodic, there is an unique steady-state distribution whatever the initial state is. Thus if $\mathcal{S}_{\lhd p}(\phi)$ is satisfied then it is satisfied for all initial states. To check this formula we have to see if :

$$\sum_{s \in Sat(\phi)} \Pi(s) \leq p \tag{26}$$

where $Sat(\phi) = \{x \in E \mid x \models \phi\}$. We propose to verify the formula using the upper bounds on the probability distributions Π^u. The verification can be performed as follows: if $Sat(\phi) \in \Phi_{wk}(E)$ then from the stochastic comparison of the processes:

$$\sum_{s \in Sat(\phi)} \Pi(s) \leq \sum_{s \in Sat(\phi)} \Pi^u(s) \tag{27}$$

so if $\sum_{s \in Sat(\phi)} \Pi^u(s) \leq p$ then inequality (26) is valid so the formula is satisfied. In the case of $\sum_{s \in Sat(\phi)} \Pi^l(s) > p$, then the formula is not satisfied. Otherwise, we can't conclude with bounding values.

For instance, if we are interested in verifying that loss probability in queue i is lower then p, then we search for all states $s \in E$ verifying the property $s_i = B_i$, and we need to compute $S_{\leq p}(s_i = B_i)$. As $Sat(s_i = B_i)$ represents an increasing set of $\Phi_{wk}(E)$ because $Sat(s_i = B_i) = \{x \in E \mid x \succeq s\}$, then we can verify the formula from the bounds.

4.2 Checking $\mathcal{P}_{\lhd p}(\phi_1 \mathcal{U}^{[t_1,t_2]} \phi_2)$

We propose to study the time-bounded until operator. We verify if :

$$s \models \mathcal{P}_{\lhd p}(\phi_1 \mathcal{U}^{[t_1,t_2]} \phi_2) \tag{28}$$

As an example of properties ϕ_1 and ϕ_2, we could have: ϕ_1 means that only one queue is full, and ϕ_2 means that all queues are full. The computation of this path

formula could be useful in the study of the congestion problem in a network, in order to see the impact of the overload of one queue in the overload of the whole system using the time constraint. In order to check the Until formula, we need to define a new process from $X(t)$ by partitioning the state space and making some states absorbing according to the properties ϕ_1 and ϕ_2. We suppose that $[t_1, t_2] = [0, t]$, so we study the behavior of the Markov process until we reach a state which verifies $\neg\phi_1 \vee \phi_2$. If a state verifying $\neg\phi_1 \wedge \neg\phi_2$ is reached, then the formula is not valid. So all the states which verify $\neg\phi_1 \vee \phi_2$ are made absorbing [15].

For a given property ω defined on states of the underlying Markov process $X(t)$, let $X[\omega](t)$ be a Markov process defined from $X(t)$ with infinitesimal generator $Q[\omega]$:

$$\begin{aligned} Q[\omega](x, x') &= Q(x, x') \; if \; x \not\models \omega \\ &= 0 \; otherwise \end{aligned} \tag{29}$$

where $x \not\models \omega$ means x don't verify the property ω. For the Until formula $\phi_1 \mathcal{U}^{[0,t]} \phi_2$, then the infinitesimal generator Q is transformed to $Q[\omega]$ where $\omega = \neg\phi_1 \vee \phi_2$. Let $Prob_s(\phi_1 \mathcal{U}^{[0,t]} \phi_2)$ be the probability of reaching a state verifying ϕ_2 on a path during the time interval $[0, t]$ via only ϕ_1-states, starting from the initial state s. We have:

$$s \models \mathcal{P}_{\lhd p}(\phi_1 \mathcal{U}^{[0,t]} \phi_2) \Leftrightarrow Prob_s(\phi_1 \mathcal{U}^{[0,t]} \phi_2) \lhd p \tag{30}$$

We denote by $\Pi[\omega](s, t)$ the transient probability distribution starting from state s of the process $X[\omega](t)$, and by $\Pi[\omega](s, s', t)$ the probability to be in state s' at time t starting from the initial state s. We have:

$$Prob_s(\phi_1 \mathcal{U}^{[0,t]} \phi_2) = \sum_{s' \in Sat(\phi_2)} \Pi[\omega](s, s', t) \tag{31}$$

As the transient probability distribution $\Pi[\omega](s, t)$ is very difficult to compute, we propose to use stochastic comparisons. In the former section, we have proved that: $\{X(t), t \geq 0\} \preceq_{wk} \{X^u(t), t \geq 0\}$. For the verification of the until formula, the bounding process must be transformed with the same modifications. We need to prove that: $\{X[\omega](t),\ t \geq 0\} \preceq_{wk} \{X^u[\omega](t),\ t \geq 0\}$.

$\preceq_{wk}$-Stochastic Comparison. We use theorem 1 for $\preceq_{wk}$-stochastic comparison. We begin with the monotonicity property, so we need to verify if one of the processes is monotone. We try to prove that $X^u[\omega](t)$ is $\preceq_{wk}$-monotone. Using the properties ϕ_1 and ϕ_2, the state space E can be divided into three state spaces: $Sat(\neg\phi_1 \wedge \neg\phi_2)$ (failure states), $Sat(\phi_1 \wedge \neg\phi_2)$ (inconclusive states), and $Sat(\phi_2)$ (success states) [15]. As we have explained before, for the computation of Until formula, states verifying $\omega = \neg\phi_1 \vee \phi_2$ are made absorbing, they are represented by states of sets $Sat(\neg\phi_1 \wedge \neg\phi_2)$ and $Sat(\phi_2)$.

The monotonicity property in theorem 2 corresponds to an increasing of the generator lines with the increasing of the states. So as some states will become

absorbing states, then the monotonicity condition could be not verified. Let explain the problem in more details. We study the inequality in theorem 2, for states x and y such that $x \preceq y$. We have proved previously that the process $X^u(t)$ is $\preceq_{wk}$-monotone, so Q^u verifies the inequality in theorem 2, for any states x and y such that $x \preceq y$. $Q^u[\omega]$ is defined from Q^u by making states which verify ω absorbing. For states which don't verify ω, there is no modification for the states, so as the inequality is verified for Q^u, it is also verified for $Q^u[\omega]$. Suppose that in the inequality in theorem 2 applied to Q^u, one of the state becomes absorbing.

- If state x becomes absorbing : then the inequality is still valid, because we reduce the left term as we have removed the transitions from x to the higher states and replaced by transition to a lower state which is x.
- If state y becomes absorbing, then the inequality could be not valid because it results that x could have now transitions to higher states than y, so the monotonicity could be not valid.

We can conclude for this problem that the monotonicity must verified for $Q^u[\omega]$ using the inequality in theorem 2, and it depends on the properties ϕ_1 and ϕ_2, and the order defined on the state space E.

Furthermore, the comparison of generators $Q[\omega]$ and $Q^u[\omega]$ is immediate as the generators Q and Q^u are comparable. From theorem 1, we can deduce that: $\{X[\omega](t), t \geq 0\} \preceq_{wk} \{X^u[\omega](t), t \geq 0\}$.

$$Prob_s(\phi_1 \mathcal{U}^{[0,t]} \phi_2) = \sum_{s' \in Sat(\phi_2)} \Pi[\omega](s, s', t) \tag{32}$$

The stochastic comparison of the processes induces the comparison of the transient probability distributions. So if $Sat(\phi_2) \in \Phi_{wk}(E)$, then:

$$\sum_{s' \in Sat(\phi_2)} \Pi[\omega](s, s', t) \leq \sum_{s' \in Sat(\phi_2)} \Pi^u[\omega](s, s', t) \tag{33}$$

If we want to verify on state s the following formula: $s \models \mathcal{P}_{\leq p}(\phi_1 \mathcal{U}^{[0,t]} \phi_2)$, then it is equivalent to:

$$Prob_s(\phi_1 \mathcal{U}^{[0,t]} \phi_2) \leq p \tag{34}$$

Then using the upper bound, if:

$$\sum_{s' \in Sat(\phi_2)} \Pi^u[\omega](s, s', t) \leq p \tag{35}$$

then the formula is valid on the exact process. Also, if we have defined the lower bound $X^l(t)$, then if

$$\sum_{s' \in Sat(\phi_2)} \Pi^l[\omega](s, s', t) > p \tag{36}$$

then the formula is not valid, otherwise we cannot conclude. The advantage of this comparison is that the upper bound (and the lower bound) can be computed

easily from the product of transient probability distributions of M/M/1 queues. For the upper bound we have:

$$\Pi^u[\omega](s,t) = \Pi_{i=1}^n p_i exp(tQ_i^u[\omega]) \tag{37}$$

Where Q_i^u is the infinitesimal generator of queue i, computed from Q^u. For the initial state $s = (s_1, \ldots, s_n)$, then p_i is the one dimensional probability vector of queue i such that $p_i(s_i) = 1$ otherwise 0.

Example. We suppose the following properties ϕ_1: means queue i is full, and ϕ_2 means all queues are full. The goal of this study is to verify the path formula $\mathcal{P}_{\leq p}(\phi_1 \mathcal{U}^{[0,t]} \phi_2)$, in order to evaluate the impact of the congestion of a particular queue in the congestion of all the queues.

We define $Q[\omega]$ from Q by making states which verify $\omega = \neg\phi_1 \vee \phi_2$ absorbing states, and similarly $Q^u[\omega]$ from Q^u. The state space $E = \mathbb{N}^n$ can be represented by three sets according to the properties ϕ_1 and ϕ_2:

$$E = Sat(\neg\phi_1 \wedge \neg\phi_2) \cup Sat(\phi_1 \wedge \neg\phi_2) \cup Sat(\phi_2) \tag{38}$$

where $Sat(\neg\phi_1 \wedge \neg\phi_2) = \{x \in E \mid \ x \neq (B_1, \ldots B_n)\}$, $Sat(\phi_1 \wedge \neg\phi_2) = \{x \in E \mid x_i = B_i, and\ x \neq (B_1, \ldots B_n)\}$, and $Sat(\phi_2) = \{x \in E \mid x = (B_1, \ldots B_n)\}$.

The monotonicity property is verified for $Q^u[\omega]$ in the case of states belonging to the same set (as it is true for Q^u), because states x and y are either both absorbing or not.

The problem could happen for states x and y belonging to different sets, in the case of one of the state becomes absorbing. In fact, as we have explained before, if the state y becomes absorbing. If we apply the order component by component, then we can see that states of $Sat(\phi_1 \wedge \neg\phi_2)$ are either greater or not comparable with states of $Sat(\neg\phi_1 \wedge \neg\phi_2)$ which are made absorbing. In the inequality of theorem 2 we can have only $x \in Sat(\neg\phi_1 \wedge \neg\phi_2)$ and $y \in Sat(\phi_1 \wedge \neg\phi_2)$, then the inequality is valid for $Q^u[\omega]$.

For the set $Sat(\phi_2)$, there is no problem to make the state absorbing as it is the upper state (we couldn't have a transition to an upper state).

The comparison of the two generators $Q[\omega]$ and $Q^u[\omega]$ is also verified, so we obtain the comparison of the transient distributions. As $Sat(\phi_2) \in \phi_{wk}(E)$, then we obtain for the until formula:

$$Prob_s(\phi_1 \mathcal{U}^{[0,t]} \phi_2) \leq \sum_{s' \in Sat(\phi_2)} \Pi^u[\omega](s, s', t) \tag{39}$$

and the verification could be performed from the upper bound.

5 Conclusion

This paper develops an increasing set formalism in order to apply weak orderings for the model checking of multidimensional Markov processes. We study

the steady state and path formulas of CSL logic. As an example, we present a queueing system similar to a Jackson network with finite capacity queues. We prove that the system can be bounded by independent M/M/1 queues for which the stationary and transient distributions are easily computed. The weak ordering generate the comparison of tail probability distributions, allowing the comparison of some state and transient formulas. As a future work, we consider to improve the quality of the bounds, by defining bounding systems as a set of independent sub-networks.

References

1. Aziz, A., Sanwal, K., Singhal, V., Brayton, R.: Model Checking Continuous Time Markov Chains. ACM Trans. on Comp. Logic 1(1), 162–170 (2000)
2. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.P.: Model-Checking Algorithms for Continuous Markov Chains. IEEE Transactions on Software Engineering 29(6) (June 2003)
3. Ben Mamoun, M., Pekergin, N., Younès, S.: Model Checking of Continuous Time Markov Chains by Closed-Form Bounding Distributions. In: Qest 2006 (2006)
4. Castel-Taleb, H., Mokdad, L., Pekergin, N.: Model Checking of steady-state rewards using bounding aggregations. In: 2008 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS 2008, Edinburgh, UK, June 16-18 (2008)
5. Castel-Taleb, H., Pekergin, N.: Stochastic monotonicity in queueing networks (submitted)
6. Clarke, E.M., Emerson, A., Sistla, A.P.: Automatic verification of finite-state concurrent systems using temporal logic spacifications. ACM Trans. on Programming Languages and Systems 8(2), 244–263 (1986)
7. Fourneau, J.M., Pekergin, N.: An algorithmic approach to stochastic bounds. In: Calzarossa, M.C., Tucci, S. (eds.) Performance 2002. LNCS, vol. 2459, p. 64. Springer, Heidelberg (2002)
8. Lindvall, T.: Lectures on the coupling method. Wiley series in Probability and Mathematical statistics (1992)
9. Lindvall, T.: Stochastic monotonicities in Jackson queueing networks. Prob. in the Engineering and Informational Sciences 11, 1–9 (1997)
10. Massey, W.: Stochastic orderings for Markov processes on partially ordered spaces. Mathematics of Operations Research 12(2) (May 1987)
11. Massey, W.: A family of bounds for the transient behavior of a Jackson Network. J. App. Prob. 23, 543–549 (1986)
12. Pekergin, N., Younes, S.: Stochastic model checking with stochastic comparison. In: Bravetti, M., Kloul, L., Zavattaro, G. (eds.) EPEW/WS-EM 2005. LNCS, vol. 3670, pp. 109–124. Springer, Heidelberg (2005)
13. Remke, A., Haverkort, B.R.: CSL model checking algorithms for infinite-state structured Markov chains
14. Stoyan, D.: Comparison methods for queues and other stochastics models. J. Wiley and son, Chichester (1976)
15. Younes, S.: Model Checking Stochastique par la méthode de comparaison stochastique, Thèse de doctorat de l'université de Versailles Saint Quentin (December 2008)

Numerical Method for Bounds Computations of Discrete-Time Markov Chains with Different State Spaces

Mourad Ahmane[1] and Laurent Truffet[2]

[1] SET Laboratory, University of Technology of Belfort-Montbéliard, 90010 Belfort, France
mourad.ahmane@utbm.fr
[2] Ecole des Mines de Nantes, 4 rue Alfred Kastler, BP.20722, 44307 Nantes Cedex3, France
laurent.truffet@emn.fr

Abstract. In this paper, we propose a numerical method for bounds computations of discrete-time Markov chains with different state spaces. This method is based on the necessary and sufficient conditions for the comparison of one-dimensional (also known as the point-wise comparison) of discrete-time Markov chains given in our previous work [3]. For achieving our objective, we proceed as follows. Firstly, we transform the comparison criterion under the form of a complete linear system of inequalities. Secondly, we use our implementation on Scilab software of Gamma-algorithm to determine the set of all possible bounds of a given Markov chain.

Keywords: Discrete-time Markov chains, comparison, bounds, Gamma-algorithm.

Basic Notations

- For all $m \in \mathbb{N}^*$, $\leq_m$ is the component-wise ordering of $\mathbb{R}^m$.
- Vectors are column vectors. $()^\top$ is the standard transpose operator.
- Vectors and matrices are written in Boldsymbol (i.e. vector $\boldsymbol{x}$, matrix $\boldsymbol{A}$).
- The set $\mathcal{S}_k$ denotes the set of all k-dimensional stochastic vectors.
- The set of real (resp. non-negative) $k \times n$-matrices is denoted by $\mathcal{M}_{k,n}(\mathbb{R})$ (resp. $\mathcal{M}_{k,n}(\mathbb{R}_+)$). $\mathbf{1}_k$ (resp. $\mathbf{0}_k$) denotes the k-dimensional vector with all components are 1 (resp. 0). $\mathbf{0}_{m\times k}$ is a matrix with m rows and k columns such that all its elements are 0. $\mathbf{0}_{(m\times k)}$ is the $(m \times k)$-dimensional vector with all coefficients are 0. $\boldsymbol{A}_{(m\times k)\times(n\times d)}$ define a matrix $\boldsymbol{A}$ with $(m \times k)$-row vectors and $(n \times d)$-column vectors.
- $\boldsymbol{K}_{l,.}$, $\boldsymbol{K}_{.,d}$ are the l^{th} row and d^{th} column vectors of matrix $\boldsymbol{K}$, respectively.
- If $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{M}_{m,n}(\mathbb{R})$ then $\boldsymbol{A} \leq \boldsymbol{B}$ denotes the entry-wise comparison of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 309–323, 2009.

1 Introduction

Markovian models have been proved useful in many areas of applied probability, specially in performance evaluation. In general, the exact computation of the transient/stationary state probability vector is needed. However, a standard problem in Markov modeling is the state dimensional explosion. Furthermore, Markov models for which nice computational properties like product-form property or matrix-geometric solutions apply, are rare. Hence, an exact computation can not always be achieved. This makes state space reduction techniques very attractive. These techniques provide approximate solutions except when a (weak) lumpability property holds (see Kemeney and Snell [10]). State space reduction techniques are usually efficient and work well. However, they provide no warranty on the error. To overcome such a limitation, methods for computing bounds on the performance parameters of interest may be used. When upper and lower bounds can be derived, they also provide an upper bound for the error on the estimates of the performance parameters. Standard performance parameters are interpreted as functionals of Markov chains. Thus, performance evaluation requires to deal with state space reduction policies and functional of Markov chains. In order to obtain bounds, methods for the comparison of state probability vectors of Markov chains with different state spaces are needed (e.g. see Pekergin [16], Doisy [6], Abu-Amsha and Vincent [1], Ledoux and Truffet [12], Truffet [17] and references therein). Let us mention that the matrix-based approach developed here, is inspired by earlier results of Keilson and Kester [9], Kester [8], Whitt [18], Massey [14], Li and Shaked [13], Kijima [11] and references therein. Hereafter, we give the precise formulation of the problem addressed here.

Let E and F be two finite sets with respective cardinal d and d'. Let $\boldsymbol{X} = (\boldsymbol{X}_n)_n$ be a Markov chain with state space E and transition probability matrix (t.p.m.) $\boldsymbol{A}$. Consider a second Markov chain $\boldsymbol{Y} = (\boldsymbol{Y}_n)_n$ with state space F and t.p.m $\boldsymbol{B}$. The probability distribution of the random variable $\boldsymbol{X}_n$ (resp. $\boldsymbol{Y}_n$) is denoted by $\boldsymbol{x}(n)$ (resp. $\boldsymbol{y}(n)$). The sequences $(\boldsymbol{x}(n))_n$ and $(\boldsymbol{y}(n))_n$ are the one-dimensional distributions/marginal distributions of the stochastic processes $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. They satisfy the following linear systems of difference equations:

$$\begin{cases} \boldsymbol{x}(0) \in \mathcal{S}_d, \\ \boldsymbol{x}(n) = \boldsymbol{A}\boldsymbol{x}(n-1), \quad n \geq 1, \end{cases} \tag{1}$$

and

$$\begin{cases} \boldsymbol{y}(0) \in \mathcal{S}_{d'}, \\ \boldsymbol{y}(n) = \boldsymbol{B}\boldsymbol{y}(n-1), \quad n \geq 1. \end{cases} \tag{2}$$

where $\mathcal{S}_d$ and $\mathcal{S}_{d'}$ are the sets of probability vectors on $\mathbb{R}^d$ and $\mathbb{R}^{d'}$, respectively.

Let us consider two matrices $\boldsymbol{K} \in \mathcal{M}_{m,d}(\mathbb{R})$ and $\boldsymbol{K}' \in \mathcal{M}_{m,d'}(\mathbb{R})$. We define the binary relation $\leq_{\boldsymbol{K},\boldsymbol{K}'}$ by:

$$(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{S}_d \times \mathcal{S}_{d'} : \qquad \boldsymbol{x} \leq_{\boldsymbol{K},\boldsymbol{K}'} \boldsymbol{y} \Longleftrightarrow \boldsymbol{K}\boldsymbol{x} \leq_m \boldsymbol{K}'\boldsymbol{y}. \tag{3}$$

The binary relation (3) can be seen as an extension of the notion of integral stochastic order (see e.g. Muller and Stoyan [15, Chap 2]) for discrete random variables.

We say that the Markov chains $\boldsymbol{X}$ and $\boldsymbol{Y}$ are $(\boldsymbol{K}, \boldsymbol{K}')$-comparable if the following assertion is true

$$\big(\boldsymbol{x}(0) \in \mathcal{S}_d, \boldsymbol{y}(0) \in \mathcal{S}_{d'} \text{ and } \boldsymbol{x}(0) \leq_{\boldsymbol{K},\boldsymbol{K}'} \boldsymbol{y}(0)\big) \Longrightarrow \forall n \in \mathbb{N},\ \boldsymbol{x}(n) \leq_{\boldsymbol{K},\boldsymbol{K}'} \boldsymbol{y}(n). \tag{4}$$

Let us introduce the following condition:

$$\mathsf{H}:\ \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{S}_d \times \mathcal{S}_{d'} \mid \boldsymbol{K}\boldsymbol{x} \leq_m \boldsymbol{K}'\boldsymbol{y}\} \neq \emptyset. \tag{5}$$

Noticing that condition (4) is logically equivalent to:

$$\big(\boldsymbol{x} \in \mathcal{S}_d,\ \boldsymbol{y} \in \mathcal{S}_{d'},\ \text{ and } \boldsymbol{x} \leq_{\boldsymbol{K},\boldsymbol{K}'} \boldsymbol{y}\big) \Longrightarrow \boldsymbol{A}\boldsymbol{x} \leq_{\boldsymbol{K},\boldsymbol{K}'} \boldsymbol{B}\boldsymbol{y}. \tag{6}$$

The starting point of our numerical method for bounds computations is the following Theorem which is reported in Ahmane and al. ([2], [3]). This Theorem gives necessary and sufficient conditions (NSC) for $(\boldsymbol{K}, \boldsymbol{K}')$-comparison of two Markov chains.

Theorem 1 (NSC for $(\boldsymbol{K}, \boldsymbol{K}')$-comparison)
Assume H (cf. (5)). The two Markov chains defined by (1) and (2) are said to be $(\boldsymbol{K}, \boldsymbol{K}')$-comparable if and only if there exist non-negative matrix $\boldsymbol{H} \in \mathcal{M}_{m,m}(\mathbb{R}_+)$ and vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^m$ such that:

$$\begin{cases} -\boldsymbol{u}\boldsymbol{1}_d^\top \leq \boldsymbol{H}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{A} \\ \boldsymbol{H}\boldsymbol{K}' - \boldsymbol{K}'\boldsymbol{B} \leq \boldsymbol{v}\boldsymbol{1}_{d'}^\top \\ \boldsymbol{u} + \boldsymbol{v} \leq \boldsymbol{0}_m. \end{cases} \tag{7}$$

In this paper, we propose a numerical method for bounds computations of Markov chains deducted from the algebraic criterion of $(\boldsymbol{K}, \boldsymbol{K}')$-comparison given in **Theorem 1**. This method consists to transform the algebraic criterion under the standard form of a linear system of inequalities which permits us to determine the set of all possible bounds of a given Markov chain.

We draw the reader attention that the $(\boldsymbol{K}, \boldsymbol{K}')$-comparison have two important properties: the first one is that $(\boldsymbol{K}, \boldsymbol{K}')$-comparison permit us to compare two systems with different dimensions; and the second one is that matrices $\boldsymbol{K}$ and $\boldsymbol{K}'$ are not invertible.

This paper is organized as follows. In Section 2, we recall some definitions about cones and solutions of system of inequalities. Section 3 is dedicated to propose a method for the transformation of the algebraic criterion of $(\boldsymbol{K}, \boldsymbol{K}')$-comparison of **Theorem 1** under the form of a linear system of inequalities $\boldsymbol{M}\boldsymbol{x} \leq \boldsymbol{b}$. In section 4, we give the method for solving $\boldsymbol{M}\boldsymbol{x} \leq \boldsymbol{b}$. Section 5 is devoted to present the Gamma-algorithm implemented on Scilab, that will permit us to solve this linear system of inequalities. Illustrative example is given in Section 6. Finally, we conclude in Section 7.

2 Background

In this section we remind the reader about some elementary concepts about cones and the solution of system of inequalities.

2.1 Polyhedral Convex Cones

Polyhedral convex cones play an important role in dealing with systems of inequalities.

Definition 1. *Let* $\boldsymbol{A}$ *be a matrix, and* $\{\boldsymbol{a}_1, ..., \boldsymbol{a}_m\}$ *be the set of its column vectors. The set* $\boldsymbol{A}_\pi$ *given by:* $\boldsymbol{A}_\pi = \{\boldsymbol{x} \in \mathbb{E}^n | \boldsymbol{x} = \pi_1 \boldsymbol{a}_1 + ... + \pi_m \boldsymbol{a}_m, \pi_i \geq 0; i = 1, ..., m\}$ *of all nonnegative linear combinations of the column vectors of* $\boldsymbol{A}$ *is known as the finitely generated cone, the polyhedral convex cone or simply the cone generated by* $\boldsymbol{A}$*. The vectors* $\{\boldsymbol{a}_1, ..., \boldsymbol{a}_m\}$ *are called 'cone generators'.*

Similarly, we shall denote $\boldsymbol{A}_\rho$ as the linear space generated by columns of $\boldsymbol{A}$.

Definition 2 (standard form of a cone). *Let* $\boldsymbol{A}_\pi$ *be a cone. Its generators* $\{\boldsymbol{a}_1, ..., \boldsymbol{a}_m\}$ *can be classify into two groups:*

1. *The generators whose opposite vectors belong to the cone, that is,* $\boldsymbol{C} \equiv \{\boldsymbol{a}_i | -\boldsymbol{a}_i \in \boldsymbol{A}_\pi\}$.
2. *The generators whose opposite vectors do not belong to the cone, that is,* $\boldsymbol{D} \equiv \{\boldsymbol{a}_i | -\boldsymbol{a}_i \notin \boldsymbol{A}_\pi\}$.

Thus, the cone can be expressed in the following form:

$$\boldsymbol{A}_\pi \equiv (\boldsymbol{C} \mid -\boldsymbol{C} \mid \boldsymbol{D})_\pi \equiv \boldsymbol{C}_\rho + \boldsymbol{D}_\pi,$$

which is known as the standard form of a cone. The standard form of a cone distinguishes between its linear space part $\boldsymbol{C}_\rho$ *and its proper cone part* $\boldsymbol{D}_\pi$.

2.2 Dual Cones

Definition 3 (Non-positive dual or polar cone). *Let* $\boldsymbol{A}_\pi$ *be a cone in* $\mathbb{E}^n$*, with generators* $\{\boldsymbol{a}_1, ..., \boldsymbol{a}_m\}$*. The non-positive dual or polar* $\boldsymbol{A}_\pi^p$ *(denoted with a p super-index) of* $\boldsymbol{A}_\pi$ *is defined as the set:*

$$\boldsymbol{A}_\pi^p \equiv \{\boldsymbol{v} \in \mathbb{E}^n | \boldsymbol{A}^T \boldsymbol{v} \leq \boldsymbol{0}\} \equiv \{\boldsymbol{v} \in \mathbb{E}^n | \boldsymbol{a}_i^T \boldsymbol{v} \leq 0;\ i = 1, ..., m\}.$$

Note that the dual of a cone is the set of vectors such that their products by those of the cone are non-positive.

2.3 Solutions of Systems of Inequalities

It is well known that the general solution of a system of inequalities $\boldsymbol{M}\boldsymbol{x} \leq \boldsymbol{b}$ is a polyhedron, that can be written, in its minimal form, as the sum of a linear space, polyhedral convex cone and a polytope, i.e.:

$$\boldsymbol{x} = \sum_i \rho_i v_i + \sum_j \pi_j w_j + \sum_k \lambda_k q_k, \quad \rho_i \in \mathbb{R},\ \pi_j, \lambda_k \geq 0,\ \sum_k \lambda_k = 1, \qquad (8)$$

where $v_i, w_j, q_k \in \mathbb{E}^n$. Our aim is to obtain a minimal set of generators for this polyhedron.

Algorithm1. Castillo and al. [5] proposed the following algorithm to solve $\boldsymbol{Mx} \leq \boldsymbol{b}$

Step 1. Obtain the dual cone $\mathcal{K} = \boldsymbol{V}_\rho + \boldsymbol{Z}_\pi$.
Step 2. Normalize the vectors of $\boldsymbol{Z}$ with non-null last component z_{n+1} by dividing them by z_{n+1}.
Step 3. Write $\mathcal{K}$ as $\mathcal{K} = \boldsymbol{V}_\rho + \boldsymbol{W}_\pi + \boldsymbol{Q}_\lambda$, where $\boldsymbol{W}$ and $\boldsymbol{Q}$ are the vectors in $\boldsymbol{Z}$ with null and unit last component, respectively.
Step 4. Remove the $(n+1)$ component of all vectors.

3 Transformation of $(\boldsymbol{K}, \boldsymbol{K}')$-Comparison into $\boldsymbol{Mx} \leq \boldsymbol{b}$

In this section we transform the $(\boldsymbol{K}, \boldsymbol{K}')$-comparison of **Theorem 1** p. 311 under the form of the classical linear system of inequalities $\boldsymbol{Mx} \leq \boldsymbol{b}$, where matrix $\boldsymbol{M}$, vectors $\boldsymbol{x}$ and $\boldsymbol{b}$ to be determined. For this, equation (7) becomes:

$$\begin{cases} -\boldsymbol{HK} - \boldsymbol{u}\mathbf{1}_d^\top & \leq & -\boldsymbol{KA} \\ \boldsymbol{HK}' - \boldsymbol{K}'\boldsymbol{B} - \boldsymbol{v}\mathbf{1}_{d'}^\top & \leq & \mathbf{0} \\ \boldsymbol{u} + \boldsymbol{v} & \leq_m & \mathbf{0}_m. \end{cases} \tag{9}$$

Also from **Theorem 1**, matrix $\boldsymbol{H}$ must be non-negative, then we add for (9) the following inequality:

$$\boldsymbol{H} \geq \mathbf{0}_{m\times m} \Longleftrightarrow -\boldsymbol{H} \leq \mathbf{0}_{m\times m}. \tag{10}$$

Since we treat Markov chains, the lower bound (matrix $\boldsymbol{A}$) and the upper bound (matrix $\boldsymbol{B}$) must be column stochastic vectors. Then if we look to determine the unknown matrix $\boldsymbol{A}$ (lower bound), we need adding to (9) the following supplementary inequalities:

$$\begin{cases} \sum_{i=1}^{i=d} A_{(i,j)} = 1, j = 1, ..., d \\ -\boldsymbol{A} \leq \mathbf{0}_{d\times d} \end{cases}, \quad \text{which is equivalent to:}$$

$$\begin{cases} -\sum_{i=1}^{i=d} A_{(i,j)} \leq -1 \\ \sum_{i=1}^{i=d} A_{(i,j)} \leq 1 \\ -\boldsymbol{A} \leq \mathbf{0}_{d\times d}, \end{cases} \tag{11}$$

and therefore in the case where $\boldsymbol{B}$ is unknown (upper bound), we add:

$$\begin{cases} -\sum_{i=1}^{i=d'} B_{(i,j)} \leq -1;\ j = 1, ..., d' \\ \sum_{i=1}^{i=d'} B_{(i,j)} \leq 1 \\ -\boldsymbol{B} \leq \mathbf{0}_{d'\times d'}. \end{cases} \tag{12}$$

Finally, by assembling all the inequalities previously defined by (9, 10, 11), we obtain in the case where $\boldsymbol{A}$ is unknown the following system of inequalities:

$$\begin{cases} -\boldsymbol{HK} - \boldsymbol{u}\mathbf{1}_d^\top - \boldsymbol{KA} \leq \mathbf{0}_{m\times d} \\ \boldsymbol{HK'} - \boldsymbol{v}\mathbf{1}_{d'}^\top \leq \boldsymbol{K'B} \\ \boldsymbol{u}+\boldsymbol{v} \leq \mathbf{0}_m \\ -\boldsymbol{H} \leq \mathbf{0}_{m\times m} \\ -\boldsymbol{A} \leq \mathbf{0}_{d\times d} \\ -\sum_{i=1}^{i=d} A_{(i,j)} \leq -\mathbf{1}_d \\ \sum_{i=1}^{i=d} A_{(i,j)} \leq \mathbf{1}_d, \end{cases} \tag{13}$$

and in the case where $\boldsymbol{B}$ is unknown, by assembling (9, 10, 12) we obtain the following system of inequalities:

$$\begin{cases} -\boldsymbol{HK} - \boldsymbol{u}\mathbf{1}_d^\top \leq -\boldsymbol{KA} \\ \boldsymbol{HK'} - \boldsymbol{K'B} - \boldsymbol{v}\mathbf{1}_{d'}^\top \leq \mathbf{0}_{m\times d'} \\ \boldsymbol{u}+\boldsymbol{v} \leq \mathbf{0}_m \\ -\boldsymbol{H} \leq \mathbf{0}_{m\times m} \\ -\boldsymbol{B} \leq \mathbf{0}_{d'\times d'} \\ -\sum_{i=1}^{i=d'} B_{(i,j)} \leq -\mathbf{1}_{d'} \\ \sum_{i=1}^{i=d'} B_{(i,j)} \leq \mathbf{1}_{d'}. \end{cases} \tag{14}$$

The objective now is to make Systems (13) and (14) under the form of $\boldsymbol{Mx} \leq \boldsymbol{b}$. Then, for (13) the column vector $\boldsymbol{b}$ is defined in this case by:

$$\boldsymbol{b}^T = (\mathbf{0}_{(m\times d)}, (\boldsymbol{K'B})_{.,1}, \ldots, (\boldsymbol{K'B})_{.,d'}, \mathbf{0}_{m+(m\times m)+(d\times d)}, -\mathbf{1}_d, \mathbf{1}_d). \tag{15}$$

And for system (14), the column vector $\boldsymbol{b}$ is defined in this case by:

$$\boldsymbol{b}^T = ((-\boldsymbol{KA})_{.,1}, \ldots, (-\boldsymbol{KA})_{.,d}, \mathbf{0}_{(m\times d')+m+(m\times m)+(d'\times d')}, -\mathbf{1}_{d'}, \mathbf{1}_{d'}). \tag{16}$$

It remains now to determine the different elements constituting matrix $\boldsymbol{M}$ and vector $\boldsymbol{x}$ for system (13) (resp. (14)). For that, we proceed as follows. We put all the constant elements (without the elemets constituting vectors $\boldsymbol{b}$ defined previously) of system (13) (resp. (14)) in matrix $\boldsymbol{M}$ and the unknown elements in vector $\boldsymbol{x}$. However, in the two systems of inequalities previously defined by (13) and (14), we remark that in certain matrix products of the same inequality, unknown matrices are sometimes on the left and sometimes on the right. More precisely, let us look for example the 1^{st} inequality of system (13): $-\boldsymbol{HK} - \boldsymbol{u}\mathbf{1}_d^\top - \boldsymbol{KA} \leq \mathbf{0}_{m\times d}$, we note that the unknown matrices $\boldsymbol{H}$ and $\boldsymbol{A}$ are on the left and on the right of matrix $\boldsymbol{K}$, respectively. We recall that matrices $\boldsymbol{K}$ and $\boldsymbol{K'}$ are not invertible, then we recover the problem due to the non-commutability of the matrix product because in $\boldsymbol{Mx} \leq \boldsymbol{b}$, vector $\boldsymbol{x}$ of unknown elements is only on the right of matrix $\boldsymbol{M}$.

In order to solve the non-commutability problem, we propose a method which permits us to put the constant elements (without the elemets constituting vectors $\boldsymbol{b}$ defined previously) of system (13) (resp. (14)) in matrix $\boldsymbol{M}$ and the unknown elements in vector $\boldsymbol{x}$. First, we remark that matrix $\boldsymbol{M}$ and the column vector $\boldsymbol{b}$ have the same number of rows and that equal to $l = ((m \times d) + (m \times d') + m + (m \times m) + (d \times d) + (d \times 2))$ for (13) (resp. $l = ((m \times d) + (m \times d') + m + (m \times m) + (d' \times d') + (d' \times 2))$ for (14)).

Let us consider for example the 1^{st} inequality of system (13): $-\boldsymbol{HK} - \boldsymbol{u}\mathbf{1}_d^\top - \boldsymbol{KA} \leq \mathbf{0}_{m \times d}$, we remark that

$$(\boldsymbol{HK})_{i,j} = \boldsymbol{H}_{i,.}\boldsymbol{K}_{.,j} = \boldsymbol{K}_{.,j}^T \boldsymbol{H}_{i,.}^T. \tag{17}$$

In order to illustrate, let us consider a simple case where $m = d = d' = 2$. The matrix product $\boldsymbol{HK}$ is given as follows:

$$\begin{pmatrix} H_{1,1}K_{1,1} + H_{1,2}K_{2,1} & H_{1,1}K_{1,2} + H_{1,2}K_{2,2} \\ H_{2,1}K_{1,1} + H_{2,2}K_{2,1} & H_{2,1}K_{1,2} + H_{2,2}K_{2,2} \end{pmatrix},$$

which can be put under the form of a vector as follows:

$$\begin{pmatrix} H_{1,1}K_{1,1} + H_{1,2}K_{2,1} \\ H_{2,1}K_{1,1} + H_{2,2}K_{2,1} \\ H_{1,1}K_{1,2} + H_{1,2}K_{2,2} \\ H_{2,1}K_{1,2} + H_{2,2}K_{2,2} \end{pmatrix}. \tag{18}$$

According to equation (17), equation (18) is equal to:

$$\begin{pmatrix} K_{1,1} & K_{2,1} & 0 & 0 \\ 0 & 0 & K_{1,1} & K_{2,1} \\ K_{1,2} & K_{2,2} & 0 & 0 \\ 0 & 0 & K_{1,2} & K_{2,2} \end{pmatrix} \begin{pmatrix} H_{1,1} \\ H_{1*,2} \\ H_{2,1} \\ H_{2,2} \end{pmatrix}. \tag{19}$$

As seen peviously in equation (19), we have just to make the transposes of each column of matrix $\boldsymbol{K}$ in matrix $\boldsymbol{M}$.

Now, the non-commutability problem of the matrix product is resolved. We can now define in each case (system (13) or (14)) the different elements constituting matrix $\boldsymbol{M}$ and vector $\boldsymbol{x}$.

Then, in the case where $\boldsymbol{A}$ is unknown, we define the column vector $\boldsymbol{x}$ for system (13) as follows:

$$\boldsymbol{x}^T = (\boldsymbol{H}_{1,.}^T, \ldots, \boldsymbol{H}_{m,.}^T, u_1, \ldots, u_m, v_1, \ldots, v_m, \boldsymbol{A}_{.,1}, \ldots, \boldsymbol{A}_{.,d}),$$

and reciprocally in the case where $\boldsymbol{B}$ is unknown, the column vector $\boldsymbol{x}$ for system (14) is:

$$\boldsymbol{x}^T = (\boldsymbol{H}_{1,.}^T, \ldots, \boldsymbol{H}_{m,.}^T, u_1, \ldots, u_m, v_1, \ldots, v_m, \boldsymbol{B}_{.,1}, \ldots, \boldsymbol{B}_{.,d'}).$$

As seen previously, the vectors $\boldsymbol{x}$ are defined in both cases. It remains only to define the element constituting matrix $\boldsymbol{M}$ in each case. Let $l = (m \times d) + (m \times$

$d') + m + (m \times m) + (d \times d) + (d \times 2)$ for (13) (resp. $l = (m \times d) + (m \times d') + m + (m \times m) + (d' \times d') + (d' \times 2)$ for (14)) be the number of row vectors of $\boldsymbol{M}$, we take $\boldsymbol{M_H} \in \mathcal{M}_{l,m\times m}(\mathbb{R})$, $\boldsymbol{M_u} \in \mathcal{M}_{l,m}(\mathbb{R})$, $\boldsymbol{M_v} \in \mathcal{M}_{l,m}(\mathbb{R})$, $\boldsymbol{M_A} \in \mathcal{M}_{l,d\times d}(\mathbb{R})$ and $\boldsymbol{M_B} \in \mathcal{M}_{l,d'\times d'}(\mathbb{R})$. For each system (13) or (14), we define matrix $\boldsymbol{M}$ as follows:

– if matrix $\boldsymbol{A}$ is unknown, for system (13) we have

$$\boldsymbol{M} = (\boldsymbol{M_H}|\boldsymbol{M_u}|\boldsymbol{M_v}|\boldsymbol{M_A})$$

– if matrix $\boldsymbol{B}$ is unknown, for system (14) we have

$$\boldsymbol{M} = (\boldsymbol{M_H}|\boldsymbol{M_u}|\boldsymbol{M_v}|\boldsymbol{M_B})$$

Then, in each case we have the following matrices:

In the case where $\boldsymbol{A}$ is unknown

$$\boldsymbol{M_H} = \begin{pmatrix}
-\boldsymbol{K}_{.,1} & 0 & \dots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \dots & 0 & -\boldsymbol{K}_{.,1} \\
\multicolumn{4}{c}{\vdots} \\
-\boldsymbol{K}_{.,d} & 0 & \dots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \dots & 0 & -\boldsymbol{K}_{.,d} \\
\boldsymbol{K'}_{.,1} & 0 & \dots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \dots & 0 & \boldsymbol{K'}_{.,1} \\
\multicolumn{4}{c}{\vdots} \\
\boldsymbol{K'}_{.,d} & 0 & \dots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \dots & 0 & \boldsymbol{K'}_{.,d'} \\
\multicolumn{4}{c}{\boldsymbol{0}_{m\times(m\times m)}} \\
\multicolumn{4}{c}{-\boldsymbol{I}_{(m\times m)}} \\
\multicolumn{4}{c}{\boldsymbol{0}_{(d\times d)\times(m\times m)}} \\
\multicolumn{4}{c}{\boldsymbol{0}_{(d\times 2)\times(m\times m)}}
\end{pmatrix}$$

In the case where $\boldsymbol{B}$ is unknown

$$\boldsymbol{M_H} = \begin{pmatrix}
-\boldsymbol{K}_{.,1} & 0 & \dots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \dots & 0 & -\boldsymbol{K}_{.,1} \\
\multicolumn{4}{c}{\vdots} \\
-\boldsymbol{K}_{.,d} & 0 & \dots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \dots & 0 & -\boldsymbol{K}_{.,d} \\
\boldsymbol{K'}_{.,1} & 0 & \dots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \dots & 0 & \boldsymbol{K'}_{.,1} \\
\multicolumn{4}{c}{\vdots} \\
\boldsymbol{K'}_{.,d'} & 0 & \dots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \dots & 0 & \boldsymbol{K'}_{.,d'} \\
\multicolumn{4}{c}{\boldsymbol{0}_{m\times(m\times m)}} \\
\multicolumn{4}{c}{-\boldsymbol{I}_{(m\times m)}} \\
\multicolumn{4}{c}{\boldsymbol{0}_{(d'\times d')\times(m\times m)}} \\
\multicolumn{4}{c}{\boldsymbol{0}_{(d'\times 2)\times(m\times m)}}
\end{pmatrix}$$

In the case where $\boldsymbol{A}$ is unknown

$$\boldsymbol{M_u} = \begin{pmatrix} -\boldsymbol{I}_m \\ \vdots \\ -\boldsymbol{I}_m \\ \boldsymbol{0}_{(m\times d')\times m} \\ \boldsymbol{I}_m \\ \boldsymbol{0}_{(m\times m)\times m} \\ \boldsymbol{0}_{(d\times d)\times m} \\ \boldsymbol{0}_{(d\times 2)\times m} \end{pmatrix}$$

In the case where $\boldsymbol{B}$ is unknown

$$\boldsymbol{M_u} = \begin{pmatrix} -\boldsymbol{I}_m \\ \vdots \\ -\boldsymbol{I}_m \\ \boldsymbol{0}_{(m\times d')\times m} \\ \boldsymbol{I}_m \\ \boldsymbol{0}_{(m\times m)\times m} \\ \boldsymbol{0}_{(d'\times d')\times m} \\ \boldsymbol{0}_{(d'\times 2)\times m} \end{pmatrix}$$

In the case where $\boldsymbol{A}$ is unknown

$$\boldsymbol{M_v} = \begin{pmatrix} \boldsymbol{0}_{(m\times d)\times m} \\ -\boldsymbol{I}_m \\ \vdots \\ -\boldsymbol{I}_m \\ \boldsymbol{I}_m \\ \boldsymbol{0}_{(m\times m)\times m} \\ \boldsymbol{0}_{(d\times d)\times m} \\ \boldsymbol{0}_{(d\times 2)\times m} \end{pmatrix}$$

In the case where $\boldsymbol{B}$ is unknown

$$\boldsymbol{M_v} = \begin{pmatrix} \boldsymbol{0}_{(m\times d)\times m} \\ -\boldsymbol{I}_m \\ \vdots \\ -\boldsymbol{I}_m \\ \boldsymbol{I}_m \\ \boldsymbol{0}_{(m\times m)\times m} \\ \boldsymbol{0}_{(d'\times d')\times m} \\ \boldsymbol{0}_{(d'\times 2)\times m} \end{pmatrix}$$

In the case where $\boldsymbol{A}$ is unknown

$$\boldsymbol{M_A} = \begin{pmatrix} \boldsymbol{K} & \boldsymbol{0}_{m\times d} & \dots & \boldsymbol{0}_{m\times d} \\ \boldsymbol{0}_{m\times d} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0}_{m\times d} \\ \boldsymbol{0}_{m\times d} & \dots & \boldsymbol{0}_{m,d} & \boldsymbol{K} \\ \boldsymbol{0}_{(m\times d')\times(d\times d)} & & & \\ \boldsymbol{0}_{m\times(d\times d)} & & & \\ \boldsymbol{0}_{(m\times m)\times(d\times d)} & & & \\ -\boldsymbol{I}_{(d\times d)} & & & \\ -\boldsymbol{1}_d^T & \boldsymbol{0}_d^T & \dots & \boldsymbol{0}_d^T \\ \boldsymbol{0}_d^T & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0}_d^T \\ \boldsymbol{0}_d^T & \dots & \boldsymbol{0}_d^T & -\boldsymbol{1}_d^T \\ \boldsymbol{1}_d^T & \boldsymbol{0}_d^T & \dots & \boldsymbol{0}_d^T \\ \boldsymbol{0}_d^T & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0}_d^T \\ \boldsymbol{0}_d^T & \dots & \boldsymbol{0}_d^T & \boldsymbol{1}_d^T \end{pmatrix}$$

In the case where $\boldsymbol{B}$ is unknown

$$\boldsymbol{M_B} = \begin{pmatrix} \boldsymbol{0}_{(m\times d)\times(d'\times d')} & & & \\ -\boldsymbol{K'} & \boldsymbol{0}_{m\times d'} & \dots & \boldsymbol{0}_{m,d'} \\ \boldsymbol{0}_{m\times d'} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0}_{m\times d'} \\ \boldsymbol{0}_{m\times d'} & \dots & \boldsymbol{0}_{m\times d'} & -\boldsymbol{K'} \\ \boldsymbol{0}_{m\times(d'\times d')} & & & \\ \boldsymbol{0}_{(m\times m)\times(d'\times d')} & & & \\ -\boldsymbol{I}_{(d'\times d')} & & & \\ -\boldsymbol{1}_{d'}^T & \boldsymbol{0}_{d'}^T & \dots & \boldsymbol{0}_{d'}^T \\ \boldsymbol{0}_{d'}^T & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0}_{d'}^T \\ \boldsymbol{0}_{d'}^T & \dots & \boldsymbol{0}_{d'}^T & -\boldsymbol{1}_{d'}^T \\ \boldsymbol{1}_{d'}^T & \boldsymbol{0}_{d'}^T & \dots & \boldsymbol{0}_{d'}^T \\ \boldsymbol{0}_{d'}^T & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0}_{d'}^T \\ \boldsymbol{0}_{d'}^T & \dots & \boldsymbol{0}_{d'}^T & \boldsymbol{1}_{d'}^T \end{pmatrix}$$

4 Resolution of the System of Inequalities $\boldsymbol{Mx} \leq \boldsymbol{b}$

After having defined in Section 3 all the elements $\boldsymbol{M}$, $\boldsymbol{x}$ and $\boldsymbol{b}$ of $\boldsymbol{Mx} \leq \boldsymbol{b}$, the objective now is to enumerate all possible solutions in $\boldsymbol{x}$ for $\boldsymbol{Mx} \leq \boldsymbol{b}$ by using Gamma-algorithm which will be defined in Section 5 p. 319.

Let us consider l, k the rows and columns numbers of matrix $\boldsymbol{M}$ respectively. The system of inequalities $\boldsymbol{Mx} \leq \boldsymbol{b}$ can be written as follows:

$$\begin{cases} M_{1,1}x_1 + M_{1,2}x_2 \ldots + M_{1,n}x_n \leq b_1 \\ M_{2,1}x_1 + M_{2,2}x_2 \ldots + M_{2,n}x_n \leq b_2 \\ \ldots \quad\quad \ldots \quad\quad \ldots \quad\quad \ldots \quad\quad \leq \ldots \\ M_{m,1}x_1 + M_{m,2}x_2 \ldots + M_{m,n}x_n \leq b_m. \end{cases} \tag{20}$$

Before using Gamma-algorithm we need to transform the system of inequalities (20), by using the technique proposed by Ziegler [19], to the following homogeneous system of inequalities $\boldsymbol{Px}^* \leq \boldsymbol{0}$ with adding an extra variable $x_{n+1} = 1$ as follows:

$$\begin{cases} M_{1,1}x_1 + M_{1,2}x_2 \ldots + M_{1,n}x_n - b_1x_{n+1} \leq 0 \\ M_{2,1}x_1 + M_{2,2}x_2 \ldots + M_{2,n}x_n - b_2x_{n+1} \leq 0 \\ \ldots \quad\quad \ldots \quad\quad \ldots \quad\quad \ldots \quad\quad \ldots \quad\quad \leq \ldots \\ M_{m,1}x_1 + M_{m,2}x_2 \ldots + M_{m,n}x_n - b_mx_{n+1} \leq 0 \\ -x_{n+1} \leq 0 \\ x_{n+1} = 1 \end{cases} \tag{21}$$

which can be written also as:

$$\begin{cases} (M_{1,1}, \ldots, M_{1,n}, -b_1)(x_1, \ldots, x_n, x_{n+1})^T \leq 0 \\ (M_{2,1}, \ldots, M_{2,n}, -b_2)(x_1, \ldots, x_n, x_{n+1})^T \leq 0 \\ \ldots \leq \ldots \\ (M_{m,1}, \ldots, M_{m,n}, -b_m)(x_1, \ldots, x_n, x_{n+1})^T \leq 0 \\ -x_{n+1} \leq 0 \\ x_{n+1} = 1 \end{cases} \tag{22}$$

The system of inequalities (22) shows that $(x_1, \ldots, x_n, x_{n+1})$ is included in the dual cone of the cone generated by all the vectors:

$$\{(M_{1,1}, \ldots, M_{1,n}, -b_1), (M_{2,1}, \ldots, M_{2,n}, -b_2), \ldots, (M_{m,1}, \ldots, M_{m,n}, -b_m), (0, 0, \ldots, 0, -1)\}.$$

The system of inequalities (22) can be written as follows:

$$\begin{cases} \left(\begin{array}{c|c} \boldsymbol{M} & -\boldsymbol{b} \\ \hline \boldsymbol{0} & -1 \end{array}\right) \begin{pmatrix} \boldsymbol{x} \\ x_{n+1} \end{pmatrix} \leq 0 \\ x_{n+1} = 1 \end{cases} \quad \text{where } \boldsymbol{P} = \left(\begin{array}{c|c} \boldsymbol{M} & -\boldsymbol{b} \\ \hline \boldsymbol{0} & -1 \end{array}\right), \boldsymbol{x}^* = \begin{pmatrix} \boldsymbol{x} \\ x_{n+1} \end{pmatrix}. \tag{23}$$

The advantage of this method consists firstly to solve $\boldsymbol{Px}^* \leq \boldsymbol{0}$ and secondly to force the supplementary constraint x_{n+1} to be equal to 1. The set of solutions of (23) is the dual of the cone generated by the row vectors of $(\boldsymbol{M}| - \boldsymbol{b})$.

5 Gamma-Algorithm

Introduced by Jubete [7] and developed by Castillo and al. [5], this algorithm has for main goal to spread several applications of linear algebra to the cases of polyhedral convex cones and to linear systems of inequalities. Its main applications are:

- determine if a vector belongs or not to the cone,
- get the minimal representation of a cone and its dual under the form of its linear space and its pointed cone,
- get the intersection of two cones,
- determine the compatibility of a linear system of inequalities,
- solve a linear system of inequalities.

Note that the complexity of the Gamma-algorithm is polynomial, but the complexity of the bounds computations strongly depends on the type of the starting cone.

5.1 Implementation on Scilab

1- Function *gammasepar*
As given in **Definition 2** p. 312, this function permit to separate a generators of a given cone into two groups. It is implemented on Scilab software as follows (we use the transpose of matrix $\boldsymbol{P}$ because the set of solutions of (23) is the dual of the cone generated by the **row vectors** of $(\boldsymbol{M}| - \boldsymbol{b})$). Taking $\boldsymbol{P}^T = \boldsymbol{R}$:

Function $[\boldsymbol{G}, \boldsymbol{J}] = gammasepar(\boldsymbol{R})$
Input parameters:
- $\boldsymbol{R}$: Matrix of all generating vectors of the polyhedral convex cone.
Output parameters:
- $\boldsymbol{G}$: Matrix of all generating vectors whose opposite belong in $\boldsymbol{R}$.
- $\boldsymbol{J}$: Matrix of all generating vectors whose opposite do not belong in R.

The steps of scilab algorithm of *gammasepar* function is given in Appendix section, p. 323.

2- Function *gammalgo*
As given in **Definition 3** p. 312, this function permit to obtain the dual of a given cone. It is implemented on scilab as follows:

Function $[\boldsymbol{V}, \boldsymbol{Z}] = gammalgo(\boldsymbol{G}, \boldsymbol{J}, nbiter)$
Input parameters:
- $\boldsymbol{G}$ and $\boldsymbol{J}$ are defined above.
- $nbiter$: number of vectors of $\boldsymbol{R}$ (i.e. number of iteration).
Output parameters:
- $\boldsymbol{V}$: Matrix of all generating vectors of the dual of the linear space.
- $\boldsymbol{Z}$: Matrix of all generating vectors of the dual of the polyhedral convex cone and the polytope.

After obtaining $\boldsymbol{Z}$, we will normalize its column-vectors with non-null last component z_{n+1} by dividing them by z_{n+1} as showing in **Algorithm1** p. 313.

The steps of Scilab algorithm of *gammasepar* function is given in Appendix section, p. 323.

6 Illustrative Example

Considering the following 3×3 dimensional transition probability matrix of Markov chain:

$$\boldsymbol{A} = \begin{pmatrix} b_0 & b_0 & 0 \\ b_1 & b_1 & b_0 \\ b_2 & b_2 & b_1 + b_2 \end{pmatrix} = \begin{pmatrix} 0.4 & 0.4 & 0 \\ 0.3 & 0.3 & 0.4 \\ 0.3 & 0.3 & 0.6 \end{pmatrix}.$$

Trying now to obtain an upper bound $\boldsymbol{B}$ of size 2×2 for the matrix $\boldsymbol{A}$ in the sens of the binary relation $\leq_{\boldsymbol{K},\boldsymbol{K}'}$ defined by the two following matrices:

$$\boldsymbol{K} = \begin{pmatrix} 1 & 2 & 3 \\ -1 & -2 & -3 \\ 1 & 0 & 0 \end{pmatrix}; \boldsymbol{K}' = \begin{pmatrix} -1 & -1 \\ 0 & 0 \\ -1 & -2 \end{pmatrix}.$$

After having put our example under the form of equation (23), we use the Scilab algorithm of *gammasepar* function, we obtain matrices $\boldsymbol{G}$ and $\boldsymbol{J}$ as defined in subsection 5.1 p. 319. For saving space, we omit the details. By using of Scilab algorithm of *gammalgo* function given in subsection 5.1 p. 319, we generate many vectors. Here, we are interested to find all possible bounds $\boldsymbol{B}$. In all the vectors generated by gammalgo (the size of each column vector is fifteen), their last four components $(B_{1,1}, B_{2,1}, B_{1,2}, B_{2,2})$ are equal to the last four components of the five vectors defined above. Then the set of all solutions in $\boldsymbol{x}$ (see assertion (8)) is given by:

$$\boldsymbol{x} = \begin{pmatrix} H_{1,1} \\ H_{1,2} \\ H_{1,3} \\ H_{2,1} \\ H_{2,2} \\ H_{2,3} \\ H_{3,1} \\ H_{3,2} \\ H_{3,3} \\ u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \\ B_{1,1} \\ B_{2,1} \\ B_{1,2} \\ B_{2,2} \end{pmatrix} = \pi_1 \begin{pmatrix} 0.33 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.67 \\ 0 \\ 0 \\ 0.67 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \lambda_1 \begin{pmatrix} 2.16 \\ 1.43 \\ 0.73 \\ 0.3 \\ 0.3 \\ 0 \\ 2.4 \\ 2.4 \\ 0 \\ 0.43 \\ -1.9 \\ 0.4 \\ -0.43 \\ -0.3 \\ -0.4 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1.45 \\ 0 \\ 1.45 \\ 0 \\ 0.73 \\ 0 \\ 0.7 \\ 0 \\ 0 \\ -1 \\ 0 \\ -0.3 \\ 1 \\ 0 \\ 0.3 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \lambda_3 \begin{pmatrix} 2.16 \\ 1.43 \\ 0 \\ 0.3 \\ 0.3 \\ 0 \\ 0.7 \\ 0 \\ 0.7 \\ 1.16 \\ -1.9 \\ -1 \\ -1.16 \\ -0.3 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \lambda_4 \begin{pmatrix} 2.16 \\ 1.43 \\ 0.73 \\ 0.3 \\ 0.3 \\ 0 \\ 1.2 \\ 0 \\ 0 \\ 0.43 \\ -1.9 \\ -0.8 \\ -0.43 \\ -0.3 \\ 0.8 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

where $\pi_1, \lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$ and $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$.

Taking for example $\pi_1 = 1, \lambda_1 = 0.25, \lambda_2 = 0.3, \lambda_3 = 0.45, \lambda_4 = 0$, we obtain the following upper bound $\boldsymbol{B}$:

$$\boldsymbol{B} = \begin{pmatrix} 0.75 & 0.3 \\ 0.25 & 0.7 \end{pmatrix},$$

and the following matrices of equation (7):

$$\boldsymbol{H} = \begin{pmatrix} 2.277 & 1.0010 & 1.6175 \\ 0.21 & 0.429 & 0 \\ 1.125 & 0.6 & 0.315 \end{pmatrix}; \boldsymbol{u} = \begin{pmatrix} -0.3405 \\ -1.33 \\ -0.44 \end{pmatrix}, \boldsymbol{v} = \begin{pmatrix} 0.3405 \\ -0.21 \\ 0.44 \end{pmatrix}.$$

Verification:
As showing follows, the matrices defined above verify equation (7) of **Theorem 1**, p. 311:

$$\begin{cases} -\boldsymbol{u}\mathbf{1}_3^\top = \begin{pmatrix} 0.3405 & 0.3405 & 0.3405 \\ 1.33 & 1.33 & 1.33 \\ 0.44 & 0.44 & 0.44 \end{pmatrix} \leq \boldsymbol{HK} - \boldsymbol{KA} = \begin{pmatrix} 0.9935 & 0.652 & 1.228 \\ 1.681 & 1.462 & 1.943 \\ 0.44 & 0.65 & 1.575 \end{pmatrix} \\ \boldsymbol{HK'} - \boldsymbol{K'B} = \begin{pmatrix} -2.8945 & -4.512 \\ -0.21 & -0.21 \\ -0.19 & -0.055 \end{pmatrix} \leq \boldsymbol{v}\mathbf{1}_2^\top = \begin{pmatrix} 0.3405 & 0.3405 \\ -0.21 & -0.21 \\ 0.44 & 0.44 \end{pmatrix} \\ \boldsymbol{u} + \boldsymbol{v} = \begin{pmatrix} 0 \\ -1.54 \\ 0 \end{pmatrix} \leq \mathbf{0}_3 \\ \boldsymbol{H} = \begin{pmatrix} 2.277 & 1.0010 & 1.6175 \\ 0.21 & 0.429 & 0 \\ 1.125 & 0.6 & 0.315 \end{pmatrix} \geq \mathbf{0}_{3\times 3} \end{cases}$$

7 Conclusion

In this paper, a numerical method for bounds computations of Markov chains is proposed by using Gamma-algorithm. This method is based on the necessary and sufficient conditions for the comparison of one-dimensional distributions (or point-wise comparison) of Markov chains with different state spaces. For this we proceed firstly to transform the comparison criterion under the form of a complete linear system of inequalities and secondly to use our implementation on Scilab software of Gamma-algorithm to determine the set of all possible bounds of a given Markov chain. We note that Gamma-algorithm tends to generate many vectors. That is why in the future works, we will be interested in search of new numerical methods to treat the problem.

References

1. Abu-Amsha, O., Vincent, J.M.: An Algorithm to Bound Functionals of Markov Chains with Large State Space. In: 4th INFORMS Telecommunications Conference, Boca Raton, March 8-11 (1998)
2. Ahmane, M., Ledoux, J., Truffet, L.: Criteria for the Comparison of Discrete-Time Markov Chains. In: The Thirteenth International Workshop on Matrices and Statistics in celebration of Ingram Olkin's 80th Birthday, IWMS 2004, Bedlewo, Poland, August 18-21 (2004)
3. Ahmane, M., Ledoux, J., Truffet, L.: Positive invariance of polyherons and comparison of Markov reward models with different state spaces. In: POsitive Systems: Theory and Applications (POSTA 2006). LNCIS, vol. 341, pp. 153–160. Springer, Heidelberg (2006) (invited paper)

4. Castillo, E., Jubete, F.: The -algorithm and some applications. University of Cantabria, Santander, Spain (March 2003)
5. Castillo, E., Jubete, F., Pruneda, E., Solares, C.: Obtaining simultaneous solutions of linear subsystems of inequalities and duals. Linear Algebra and its Applications 346, 131–154 (2002)
6. Doisy, M.: A Coupling Technique For Comparison of Functions of Markov Processes. Appl. Math. Decis. Sci. 4, 39–64 (2000)
7. Jubete, F.: El cono polidrico convexo. Su indenciaen el algebra lineal y la programacion no lineal. Editorial CIS, Santander, Spain (1991)
8. Kester, A.J.M.: Preservation of Cone Characterizing Properties in Markov Chains. Phd thesis, Univ. of Rochester, New York (1977)
9. Keilson, J., Kester, A.: Monotone Matrices and Monotone Markov Processes. Stochastic Process. Appl. 5, 231–241 (1977)
10. Kemeny, J.G., Snell, J.L.: Finite Markov Chains. Springer, Heidelberg (1976)
11. Kijima, M.: Markov Processes for Stochastic Modeling. Chapman-Hall, Boca Raton (1997)
12. Ledoux, J., Truffet, L.: Markovian Bounds on Functions of Finite Markov Chains. Adv. in Appl. Probab. 33, 505–519 (2001)
13. Li, H., Shaked, M.: Stochastic Convexity and Concavity of Markov Processes. Math. Oper. Res. 29, 477–493 (1994)
14. Massey, W.A.: Stochastics Orderings for Markov Processes on Partially Ordered Spaces. Math. Oper. Res. 11, 350–367 (1987)
15. Muller, A., Stoyan, D.: Comparison methods for stochastic models and risks. J. Wiley and Sons, Chichester (2002)
16. Pekergin, N.: Stochastic Performance Bounds by State Space Reduction. Performance Eval. 117, 36–37 (1999)
17. Truffet, L.: Geometrical Bounds on an Output Stream of a Queue in ATM Switch: Application to the Dimensioning Problem. In: Kouvatsos, D. (ed.) ATM Networks: Performance Modelling and Analysis, vol. 2. Chapman-Hall, London (1996)
18. Whitt, W.: Stochastic comparisons for non-Markov processes. Math. Oper. Res. 11, 608–618 (1986)
19. Ziegler, G.M.: Lectures notes en Polytopes. Graduate Texts in Mathematics. Springer, Berlin (1995)

Appendix (Scilab Algorithms)

Scilab algorithm of *gammasepar* function

Step 1: initialize the arguments of 'linpro' (for the definition of linpro, see Step 4),
Step 2: look for each column vector of a given matrix $\boldsymbol{R}$ if its opposite is positive linear combination of the column vectors of $\boldsymbol{R}$,
Step 3: if there does not exist a realizable solution, recover the error in order to prevent the stopping function,
Step 4: use the following function 'linpro' to solve the linear programming:

$$[\boldsymbol{x}, lagr, f] = \ linpro\ (\boldsymbol{p}, \boldsymbol{Ce}, \boldsymbol{be}, \boldsymbol{Ci}, \boldsymbol{Cs}, me)$$

$\boldsymbol{p}$: column vector with real coefficients of the function objective (i.e cost function). $\boldsymbol{Ce}$: matrix of constraints with real coefficients. $\boldsymbol{be}$: column vector of constraints with real coefficients. $\boldsymbol{Ci}$: column vector of the lower bounds of variables. $\boldsymbol{Cs}$: column vector of the upper bounds of variables. me: number of constraints of equalities. $\boldsymbol{x}$: column vector of solutions allowing to obtain the optimal value. $lagr$: column vector of Lagrange-multipliers. f: optimal value.
Step 5: According to Step 4, one has vector $\boldsymbol{r}$ of $\boldsymbol{R}$ in $\boldsymbol{G}$ or $\boldsymbol{J}$.

Scilab algorithm of *gammalgo* function

- Let be $\boldsymbol{V}^{(0)} = \boldsymbol{I}_n$, $\boldsymbol{Z}^{(0)} = \emptyset$, and $\boldsymbol{U}^{(0)} = (\boldsymbol{V}^{(0)} \mid \boldsymbol{Z}^{(0)})$.
- To each vector $\boldsymbol{u}_j$ of matrix $\boldsymbol{U}^{(h)}$ is associated at the h^{th} iteration the set $I_{\boldsymbol{R}(\boldsymbol{u}_j)}$ that contains the set of indices of vectors of $\boldsymbol{R}$ orthogonal to $\boldsymbol{u}_j$.

Step 1: Calculation of the scalar product $\boldsymbol{t}^h = \boldsymbol{r}_h^T \boldsymbol{U}^{(h)}$.
Step 2: Research of the pivot. Seek a column vector of $\boldsymbol{V}^{(h)}$ such that $\boldsymbol{t}^h_{pivot} \neq \boldsymbol{0}$.
Step 3: Separation of the algorithm. if a pivot were determined, continue on *Step 4*, otherwise go to *Step 5*.
Step 4: Process I.

- normalize the pivot by dividing each one of its terms by $-\boldsymbol{t}^h_{pivot}$.
- For each column vector $\boldsymbol{u}^h_j$ of $\boldsymbol{U}^{(h)} = U^h_{i,j}$ s.t. $j \neq pivot$, put $U^h_{i,j} = U^h_{i,j} + \boldsymbol{t}^h_j U^h_{j,pivot}$.
- add the index h to the set $I_{\boldsymbol{R}(\boldsymbol{u}_j)}$ for all $j \neq pivot$.
- withdraw the vector $\boldsymbol{u}^h_{pivot}$ of $\boldsymbol{V}^{(h)}$.

If $\boldsymbol{r}_h \notin \boldsymbol{G}$, add the vector $\boldsymbol{u}^h_{pivot}$ to $\boldsymbol{Z}^{(h)}$ and then go to *Step 6*.
Step 5: Process II.

- add the index h to the set $I_{\boldsymbol{R}(\boldsymbol{u}_j)}$ for all j such that $\boldsymbol{u}_j \in \boldsymbol{V}^{(h)}$.
- Let be $\boldsymbol{z}_i$ the i^{th} column vector of $\boldsymbol{Z}^{(h)} = Z^h_{i,j}$. Determine the sets:

$$I^- \equiv \{i \mid \boldsymbol{z}_i^T \boldsymbol{r}_h < 0\},\ I^+ \equiv \{i \mid \boldsymbol{z}_i^T \boldsymbol{r}_h > 0\},\ I^0 \equiv \{i \mid \boldsymbol{z}_i^T \boldsymbol{r}_h = 0\}.$$

- add the index h to the set $I_{\boldsymbol{R}(\boldsymbol{z}_i)}$ for all $i \in I^0$.

If $\boldsymbol{r}_h \in \boldsymbol{G}$, then withdraw of $\boldsymbol{Z}^{(h)}$ all vectors $\boldsymbol{z}_i$ such that $i \in I^-$or $i \in I^+$.
Otherwise, if $I^+ \equiv \emptyset$ go to *Step 6*.
Otherwise, if $I^- \equiv \emptyset$ withdraw of $\boldsymbol{Z}^{(h)}$ all $\boldsymbol{z}_i$ such that $i \in I^+$.
Otherwise, for every $i \in I^-$ and every $j \in I^+$, construct the set

$$I_{\boldsymbol{R}(Z_{i,j})} \equiv (I_{\boldsymbol{R}(\boldsymbol{z}_i)} \cap I_{\boldsymbol{R}(\boldsymbol{z}_j)}) \cup \{h\}$$

If $\nexists p \in I^-$, $q \in I^+$, $k \in I^0$ with $p \neq i$ and $q \neq j$ such that:

$$I_{\boldsymbol{R}(Z_{i,j})} \subseteq I_{\boldsymbol{R}(Z_{p,q})} \text{ and } I_{\boldsymbol{R}(Z_{i,j})} \subseteq I_{\boldsymbol{R}(\boldsymbol{z}_k)},$$

then add to $\boldsymbol{Z}^{(h)}$ for every i and j the vector $Z_{i,j} = \boldsymbol{t}^h_j \boldsymbol{z}_i - \boldsymbol{t}^h_i \boldsymbol{z}_j$, to which associate the set $I_{\boldsymbol{R}(Z_{i,j})}$.
Then withdraw of $\boldsymbol{Z}^{(h)}$ all $\boldsymbol{z}_i$ such that $i \in I^+$.
Step 6: if $h \prec m$, then $h = h + 1$ and go to *Step 1*, otherwise **END**.

Approximate Conditional Distributions of Distances between Nodes in a Two-Dimensional Sensor Network

Rodrigo S.C. Leão and Valmir C. Barbosa*

Universidade Federal do Rio de Janeiro
Programa de Engenharia de Sistemas e Computação, COPPE
Caixa Postal 68511, 21941-972 Rio de Janeiro - RJ, Brazil
rleao@cos.ufrj.br, valmir@cos.ufrj.br

Abstract. When we represent a network of sensors in Euclidean space by a graph, there are two distances between any two nodes that we may consider. One of them is the Euclidean distance. The other is the distance between the two nodes in the graph, defined to be the number of edges on a shortest path between them. In this paper, we consider a network of sensors placed uniformly at random in a two-dimensional region and study two conditional distributions related to these distances. The first is the probability distribution of distances in the graph, conditioned on Euclidean distances; the other is the probability density function associated with Euclidean distances, conditioned on distances in the graph. We study these distributions both analytically (when feasible) and by means of simulations. To the best of our knowledge, our results constitute the first of their kind and open up the possibility of discovering improved solutions to certain sensor-network problems, as for example sensor localization.

Keywords: Sensor networks, Random geometric graphs, Distance distributions.

1 Introduction

We consider a network of n sensors, each one placed at a fixed position in two-dimensional space and capable of communicating with another sensor if and only if the Euclidean distance between the two is at most R, for some constant radius $R > 0$. If δ_{ij} denotes this distance for sensors i and j, then a graph representation of the network can be obtained by letting each sensor be a node and creating an edge between any two distinct nodes i and j such that $\delta_{ij} \leq R$. Such a representation is, aside from a scale factor, equivalent to a unit disk graph [1].

Often n is a very large integer and the network is essentially unstructured, in the sense that the sensors' positions, although fixed, are generally unknown. In domains for which this holds, generalizing the graph representation in such a way that each node's position is given by random variables becomes a crucial

* Corresponding author.

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 324–338, 2009.

step, since it opens the way to the investigation of relevant distributions related to all networks that result from the same deployment process. Such a generalization, which can be done for any number of dimensions, is known as a random geometric graph [2]. Similarly to the random graphs of Erdős and Rényi [3] and related structures [4], many important properties of random geometric graphs are known, including some related to connectivity and the appearance of the giant component [5,6,7] and others more closely related to applications [8,9,10].

One curious aspect of random geometric graphs is that, if nodes are positioned uniformly at random, the expected Euclidean distance between any two nodes is a constant in the limit of very large n, depending only on the number of dimensions (two, in our case) [11]. In this case, distance-dependent analyses must necessarily couple the Euclidean distance with some other type of distance between nodes. The natural candidate is the standard graph-theoretic distance between two nodes, given by the number of edges on a shortest path between them [12]. For nodes i and j, this distance is henceforth denoted by d_{ij} and referred to simply as the distance between i and j.

Given i and j, the Euclidean distance δ_{ij} and the distance d_{ij} between the two nodes are not independent of each other, but rather interrelate in a complex way. Our goal in this paper is to explore the relationship between the two when all sensors are positioned uniformly at random in a given two-dimensional region. Specifically, for i and j two distinct nodes chosen at random, we study the probability that $d_{ij} = d$ for some integer $d > 0$, given that $\delta_{ij} = \delta$ for some real number $\delta \geq 0$. Similarly, we also study the probability density associated with $\delta_{ij} = \delta$ when $d_{ij} = d$. Our study is analytical whenever feasible, but is also computational throughout. Depending on the value of d, we are in a few cases capable of providing exact closed-form expressions, but in general what we give are approximations, either derived mathematically or inferred from simulation data exclusively.

We remark, before proceeding, that we perceive the study of distance-related distributions for random geometric graphs as having great applicability in the field of sensor networks, particularly in domains in which it is important for each sensor to have a good estimate of its location. In fact, of all possible applications that we normally envisage for sensor networks [13], network localization is crucial in all cases that require the sensed data to be tagged with reliable indications of where the data come from; it has also been shown to be important even for routing purposes [14]. So, although we do not dwell on the issue of network localization anywhere else in the paper, we now digress momentarily to clarify what we think the impact of distance-related distributions may be.

The problem of network localization has been tackled from a variety of perspectives, including rigidity-theoretic studies [15,16], approaches that are primarily algorithmic, either centralized [17,18] or distributed [19,20,21,22], and others that generalize on our assumptions by taking advantage of sensor mobility [23,22] or uneven radii [24]. In general one assumes the existence of some anchor sensors (regularly placed [25] or otherwise), for which positions are known precisely, and then the problem becomes reduced to the problem of providing,

for each of the other sensors, the Euclidean distances that separate it from three of the anchors (its tripolar coordinates with respect to those anchors, from which the sensor's position can be easily calculated [26]).

Finding a sensor's Euclidean distance to an anchor is not simple, though. Sometimes signal propagation is used for direct or indirect measurement [27, 28, 29, 30, 31, 32], but there are approaches that rely on no such techniques [33, 25, 20]. The latter include one of the most successful distributed approaches [20], which nonetheless suffers from increasing lack of accuracy as sparsity or irregularity in sensor positioning become more pronounced. The algorithm of [20] assumes, for each anchor i, that each edge on any shortest path to i is equivalent to a fixed Euclidean distance, which is estimated by i in communication with the other anchors and by simple proportionality can be used by any node to infer its Euclidean distance to i. We believe that knowledge of distance-related distributions has an important role to play in replacing this assumption and perhaps dispelling the algorithm's difficulties in the less favorable circumstances alluded to above.

We proceed in the following manner. In Section 2 we give some notation and establish the overall approach to be followed when pursuing the analytical characterization of distance-related distributions. Then in Section 3 we present the mathematical analysis of the $d = 1$ and $d = 2$ cases (already known from [34]), and in Sections 4 and 5 that of the $d = 3$ case. We continue in Section 6 with computational results related to $d \geq 1$ and close in Section 7 with some discussion and concluding remarks.

2 Overall Approach

Let i and j be two distinct, randomly chosen nodes. For $d > 0$ an integer and $\delta \geq 0$ a real number, we use $P_\delta(d)$ to denote the probability, conditioned on $\delta_{ij} = \delta$, that $d_{ij} = d$. Likewise, we use $p_d(\delta)$ to denote the probability density, conditioned on $d_{ij} = d$, associated with $\delta_{ij} = \delta$. These two quantities relate to each other in the standard way of combining integer and continuous random variables [35].

If we assume that $P_\delta(d)$ is known for all applicable values of d and δ, then it follows from Bayes' theorem that

$$p_d(\delta) = \frac{P_\delta(d)p(\delta)}{P(d)}, \tag{1}$$

where $p(\delta) > 0$ is the unconditional probability density associated with the occurrence of an Euclidean distance of δ separating two randomly chosen nodes and $P(d) > 0$ is the unconditional probability that the distance between them is d. Clearly, $P(d) = \int_{r=0}^{dR} P_r(d)p(r)\mathrm{d}r$, since $P_r(d) = 0$ for $r > dR$. Moreover, $p(r)$ is proportional to the circumference of a radius-r circle, $2\pi r$, which yields

$$p_d(\delta) = \frac{P_\delta(d)\delta}{\int_{r=0}^{dR} P_r(d)r\mathrm{d}r}. \tag{2}$$

In view of Equation (2), our approach henceforth is to concentrate on calculating $P_\delta(d)$ for all appropriate values of d and δ, and then to use the equation to obtain $p_d(\delta)$. In order to calculate $P_\delta(d)$, we fix two nodes a and b such that $\delta_{ab} = \delta$ and proceed by analyzing how the two radius-R circles (the one centered at a and the one at b) relate to each other. While doing so, we assume that the two-dimensional region containing the graph has unit area, so that the area of any of its sub-regions automatically gives the probability that it contains a randomly chosen node. We assume further that all border effects can be safely ignored (but see Section 6 for the computational setup that justifies this).

3 The Distance-1 and Distance-2 Cases

The case of $d = 1$ is straightforward, since $d_{ab} = d$ if and only if $\delta \leq R$. Consequently,

$$P_\delta(1) = \begin{cases} 1, & \text{if } \delta \leq R; \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

and, by Equation (2),

$$p_1(\delta) = \begin{cases} 2\delta/R^2, & \text{if } \delta \leq R; \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

For $d = 2$, we have $d_{ab} = d$ if and only if $\delta > R$ and at least one node k exists, with $k \notin \{a, b\}$, such that $\delta_{ak} \leq R$ and $\delta_{bk} \leq R$. The probability that this holds for a randomly chose k is given by the intersection area of the radius-R circles centered at a and b, here denoted by ρ_δ. From [26], we have

$$\rho_\delta = \begin{cases} 2R^2 \cos^{-1}(\delta/2R) - \delta\sqrt{R^2 - \delta^2/4}, & \text{if } \delta \leq 2R; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Because any node that is not a or b may, independently, belong to such intersection, we have

$$P_\delta(2) = \begin{cases} 1 - (1 - \rho_\delta)^{n-2}, & \text{if } \delta > R; \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

As for $p_2(\delta)$, it is as given by Equation (2), equaling 0 if $\delta \leq R$ or $\delta > 2R$ (we remark that a closed-form expression is obtainable also in this case, but it is too cumbersome and is for this reason omitted).

4 The Distance-3 Case: Exact Basis

The $d = 3$ case is substantially more complex than its predecessors in Section 3. We begin by noting that $d_{ab} = d$ if and only if the following three conditions hold:

C1. $\delta > R$.
C2. No node i exists such that both $\delta_{ai} \leq R$ and $\delta_{bi} \leq R$.
C3. At least one node $k \notin \{a, b\}$ exists, and for this k at least one node $\ell \notin \{a, b, k\}$, such that $\delta_{ak} \leq R$, $\delta_{k\ell} \leq R$, $\delta_{b\ell} \leq R$, $\delta_{a\ell} > R$, and finally $\delta_{bk} > R$.

For each fixed k and ℓ in Condition C3, these three conditions result from the requirement that nodes a, k, ℓ, and b, in this order, constitute a shortest path from a to b.

If we fix some node $k \notin \{a, b\}$ for which $\delta_{ak} \leq R$ and $\delta_{bk} > R$, the probability that Condition C3 is satisfied by k and a randomly chosen ℓ is a function of intersection areas of circles that varies from case to case, depending on the value of δ. There are two cases to be considered, as illustrated in Figure 1. In the first case, illustrated in part (a) of the figure, $R < \delta \leq 2R$ and node ℓ is to be found in the intersection of the radius-R circles centered at b and k, provided it is not also in the radius-R circle centered at a. The intersection area of interest results from computing the intersection area of two circles (those centered at b and k) and subtracting from it the intersection area of three circles (those centered at a, b, and k). The former of these intersection areas is given as in Equation (5), with δ_{bk} substituting for δ; as for the latter, closed-form expressions also exist, as given in [36] (it is significant, though, that the expressions for the intersection area of three circles have only been published quite recently; without them, it does not seem that the present analysis would be possible). The second case, shown in part (b) of Figure 1, is that of $2R < \delta \leq 3R$, and then the intersection area of interest is the one of the circles centered at b and k. Regardless of which case it is, we use σ_δ^k to denote the resulting area. Thus, the probability that at least one ℓ exists for fixed k is $1 - (1 - \sigma_\delta^k)^{n-3}$.

Now let $P'_\delta(3)$ be the probability that a randomly chosen k satisfies Condition C3. Let also K_δ be the region inside which such a node can be found with nonzero probability. If x_k and y_k are the Cartesian coordinates of node k, then each possible location of k inside K_δ contributes to $P'_\delta(3)$ the infinitesimal probability $[1 - (1 - \sigma_\delta^k)^{n-3}]\mathrm{d}x_k\mathrm{d}y_k$. It follows that

$$P'_\delta(3) = \int_{k \in K_\delta} [1 - (1 - \sigma_\delta^k)^{n-3}]\mathrm{d}x_k\mathrm{d}y_k. \tag{7}$$

There are three possibilities for the region K_δ, shown in parts (a) through (c) of Figure 2 as shaded regions, respectively for $R < \delta \leq R\sqrt{3}$, $R\sqrt{3} < \delta \leq 2R$, and $2R < \delta \leq 3R$. The shaded region in part (a) is delimited by four radius-R circles, the ones centered at nodes a (above and below) and b (on the right) and the ones centered at points D and E (on the left). As δ gets increased beyond $R\sqrt{3}$—and, at the threshold, point D becomes collinear with point B and node b—we move into part (b) of the figure, where the shaded region is now delimited on the left either by the radius-R circles centered at D and E or by the radius-$2R$ circle centered at b, depending on the point of common tangent between each of the radius-R circles and the radius-$2R$ circle (note that projecting either point of common tangent on the straight line that goes through a and b yields point a exactly). The next threshold leads δ beyond $2R$, and in part (c) of the figure

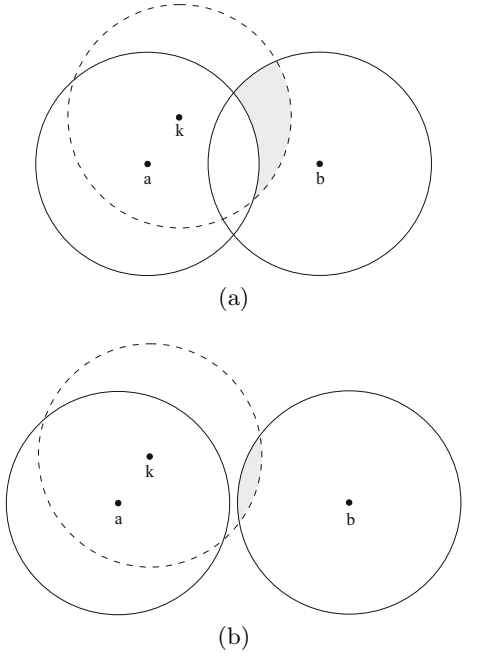


Fig. 1. Regions (shown in shades) whose areas yield the value of σ_δ^k for $R < \delta \leq 2R$ (a) and $2R < \delta \leq 3R$ (b)

the shaded region is delimited on the left by the radius-$2R$ circle centered at b, on the right by the radius-R circle centered at a.

Figure 2 is also useful in helping us obtain a more operational version of the expression for $P'_\delta(3)$, to be used in Section 6. First we establish a Cartesian coordinate system by placing its origin at node a and making the positive abscissa axis go through node b. In this system, the shaded regions in all of parts (a) through (c) of the figure are symmetrical with respect to the abscissa axis. If for each value of x_k we let $y_k^-(x_k)$ and $y_k^+(x_k)$ be, respectively, the minimum and maximum y_k values in the upper half of the shaded region for the value of δ at hand, then

$$P'_\delta(3) = 2 \int_{x_k=x_k^-}^{x_k^+} \int_{y_k=y_k^-(x_k)}^{y_k^+(x_k)} [1 - (1 - \sigma_\delta^k)^{n-3}] \mathrm{d}x_k \mathrm{d}y_k, \tag{8}$$

where x_k^- and x_k^+ bound the possible values of x_k for the given δ.

All pertinent values of x_k^- and x_k^+, as well as of $y_k^-(x_k)$ and $y_k^+(x_k)$, are given in Table 1, where δ^- and δ^+ indicate, respectively, the lower and upper limit for δ in each of the three possible cases. This table's entries make reference to the abscissae of points A, B, C, and D (respectively x_A, x_B, x_C, and x_D) and to the ordinate of point D (y_D). These are given in Table 2.

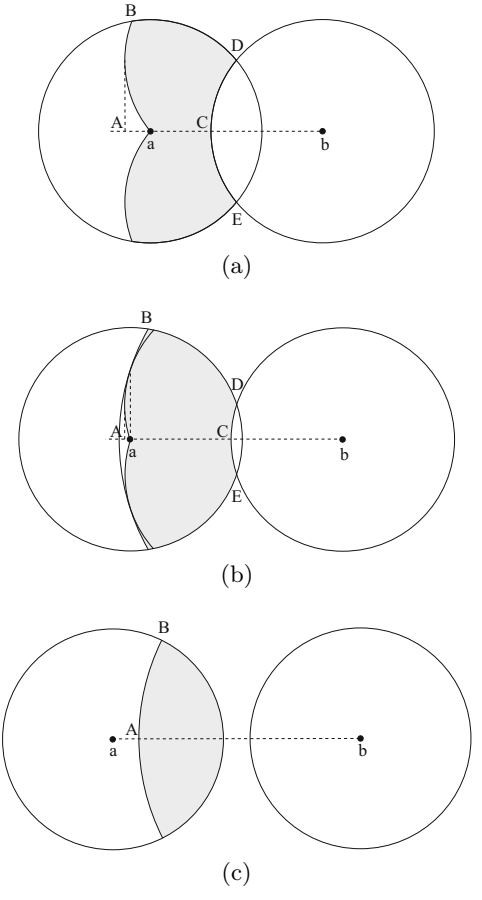


Fig. 2. Regions (shown in shades) where node k can be found with nonzero probability for $R < \delta \leq R\sqrt{3}$ (a), $R\sqrt{3} < \delta \leq 2R$ (b), and $2R < \delta \leq 3R$ (c)

5 The Distance-3 Case: Approximate Extension

Obtaining $P_\delta(3)$ from $P'_\delta(3)$ requires that we fulfill the remaining requirements set by Conditions C2 and C3 in Section 4. These are that no node exists in the intersection of the radius-R circles centered at a and b and that at least one node k exists with the properties given in Condition C3. While the probability of the former requirement is simply $(1 - \rho_\delta)^{n-2}$, expressing the probability of the latter demands that we make a careful approximation to compensate for the lack of independence of certain events with respect to one another.

For node $i \notin \{a, b\}$, let ϵ_i stand for the event that Condition C3 does not hold for $k = i$. Let also $Q_\delta(\epsilon_i)$ be the probability of ϵ_i and Q_δ the joint probability of all $n - 2$ events. Clearly, $Q_\delta(\epsilon_i) = 1 - P'_\delta(3)$ for any i and, for $\delta > R$,

Table 1. Cartesian coordinates delimiting the upper halves of shaded regions in Figure 2

δ^-	δ^+	x_k^-	x_k^+	$y_k^-(x_k)$	$y_k^+(x_k)$	Figure
R	$R\sqrt{3}$	x_A	x_B	$y_D - \sqrt{R^2 - (x_k - x_D)^2}$	$y_D + \sqrt{R^2 - (x_k - x_D)^2}$	2(a)
		x_B	0	$y_D - \sqrt{R^2 - (x_k - x_D)^2}$	$\sqrt{R^2 - x_k^2}$	
		0	x_C	0	$\sqrt{R^2 - x_k^2}$	
		x_C	x_D	$\sqrt{R^2 - (x_k - \delta)^2}$	$\sqrt{R^2 - x_k^2}$	
$R\sqrt{3}$	$2R$	x_A	0	$y_D - \sqrt{R^2 - (x_k - x_D)^2}$	$y_D + \sqrt{R^2 - (x_k - x_D)^2}$	2(b)
		0	x_B	0	$\sqrt{4R^2 - (x_k - \delta)^2}$	
		x_B	x_C	0	$\sqrt{R^2 - x_k^2}$	
		x_C	x_D	$\sqrt{R^2 - (x_k - \delta)^2}$	$\sqrt{R^2 - x_k^2}$	
$2R$	$3R$	x_A	x_B	0	$\sqrt{4R^2 - (x_k - \delta)^2}$	2(c)
		x_B	R	0	$\sqrt{R^2 - x_k^2}$	

Table 2. Cartesian coordinates used in Table 1

δ^-	δ^+	x_A	x_B	x_C	x_D	y_D	Figure
R	$R\sqrt{3}$	$\delta/2 - R$	$\left(\delta - \sqrt{3(4R^2 - \delta^2)}\right)/4$	$\delta - R$	$\delta/2$	$\sqrt{4R^2 - \delta^2}/2$	2(a)
$R\sqrt{3}$	$2R$	$\delta/2 - R$	$(\delta^2 - 3R^2)/2\delta$	$\delta - R$	$\delta/2$	$\sqrt{4R^2 - \delta^2}/2$	2(b)
$2R$	$3R$	$\delta - 2R$	$(\delta^2 - 3R^2)/2\delta$				2(c)

$P_\delta(3) = (1 - Q_\delta)(1 - \rho_\delta)^{n-2}$. Therefore, if all the $n-2$ events were independent of one another, we would have

$$Q_\delta = \prod_{i \notin \{a,b\}} Q_\delta(\epsilon_i) = [1 - P'_\delta(3)]^{n-2} \tag{9}$$

and, consequently,

$$P_\delta(3) = \begin{cases} \{1 - [1 - P'_\delta(3)]^{n-2}\}(1 - \rho_\delta)^{n-2}, & \text{if } \delta > R; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

However, once we know of a certain node i that Condition C3 does not hold for it, immediately we reassess as less likely that the condition holds for nodes in the Euclidean vicinity of i. The $n-2$ events introduced above are then not unconditionally independent of one another, although we do expect whatever degree of dependence there is to wane progressively as we move away from node i.

We build on this intuition by postulating the existence of an integer $n' < n-2$ such that the independence of the n' events not only holds but is also sufficient

to determine $P_\delta(3)$ as indicated above, provided the corresponding n' nodes are picked uniformly at random. But since this is precisely the way in which, by assumption, sensors are positioned, it suffices that any n' nodes be selected, yielding

$$P_\delta(3) = \begin{cases} \{1 - [1 - P'_\delta(3)]^{n'}\}(1 - \rho_\delta)^{n-2}, & \text{if } \delta > R; \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Similarly to the previous cases, $p_3(\delta)$ is given by Equation (2) and equals 0 if $\delta \leq R$ or $\delta > 3R$.

It remains, of course, for the value of n' to be discovered if our postulate is to be validated. We have done this empirically, by means of computer simulations, as discussed in Section 6.

6 Computational Results

In this section we present simulation results and, for $d = 1, 2, 3$, contrast them with the analytic predictions of Sections 3 through 5. The latter are obtained by numerical integration when a closed-form expression is not available (the case of $d = 3$ also requires simulations for finding n'; see below). For $d > 3$, we demonstrate that good approximations by Gaussians can be obtained.

We use $n = 1\,000$ and a circular region of unit area, therefore of radius $\sqrt{1/\pi} \approx 0.564$, for the placement of nodes. Node a is always placed at the circle's center, which has Cartesian coordinates $(0, 0)$, and all results refer to distances to a. Our choice for the value of R depends on the expected number of neighbors (or connectivity) of a node, which we denote by z and use as the main parameter. Since $z = \pi R^2 n$ for large n, choosing the value of z immediately yields the value of R to be used. We use $z = 3\pi$ and $z = 5\pi$ (though it seems that lower values of z also occur in practice [37]), which yield, respectively, $R \approx 0.055$ and $R \approx 0.071$. We note that both values of z are significantly above the phase transition that gives rise to the giant component, which happens at $z \approx 4.52$ [38]. In all our experiments, then, graphs are connected with high probability.

For each value of z, each simulation result we present is an average over 10^6 independent trials. Each trial uses a matrix of accumulators having $n - 1$ rows (one for each of the possible distance values) and $1\,000\sqrt{1/\pi}$ columns (one for each of the 0.001-wide bins into which Euclidean distances are compartmentalized). A trial consists of: placing $n - 1$ nodes uniformly at random in the circle; computing the Euclidean distance between each node and node a; computing the distances between each node and node a (this is done with Dijkstra's algorithm [39]); updating the accumulator that corresponds to each node, given its two distances. At the end of each trial, its contributions to $P_\delta(d)$ and $p_d(\delta)$ are computed, with $d = 1, 2, \ldots, n - 1$ and δ ranging through the middle points of all bins. If M is the matrix of accumulators, then these contributions are given, respectively, by $M(d, \delta) / \sum_{d'} M(d', \delta)$ and $M(d, \delta) / 0.001 \sum_{\delta'} M(d, \delta')$.

The case of $d = 3$ requires two additional simulation procedures, one for determining simulation data for $P'_\delta(3)$, the other to determine n' for use in

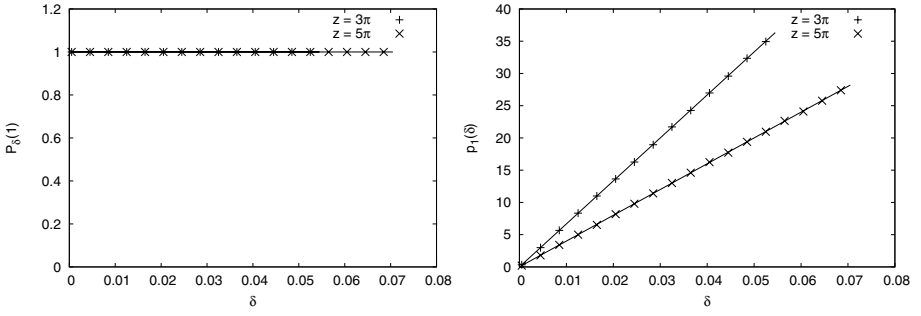


Fig. 3. $P_\delta(1)$ (a) and $p_1(\delta)$ (b). Solid lines give the analytic predictions.

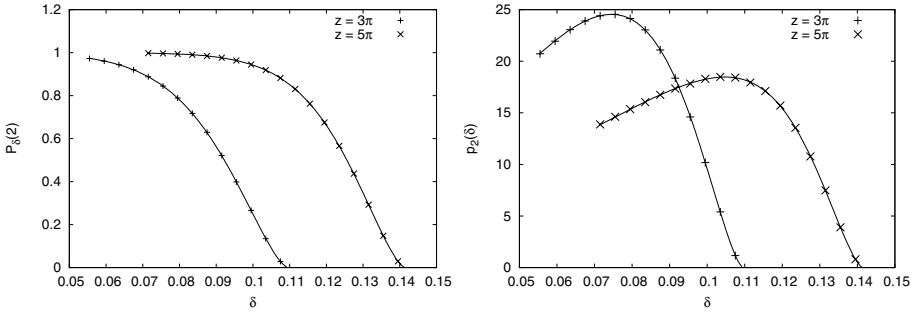


Fig. 4. $P_\delta(2)$ (a) and $p_2(\delta)$ (b). Solid lines give the analytic predictions.

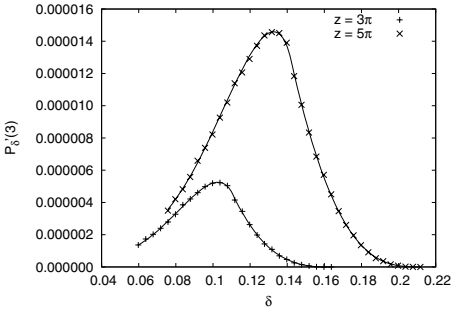


Fig. 5. $P'_\delta(3)$. Solid lines give the analytic predictions.

obtaining analytic predictions for $P_\delta(3)$. The former of these fixes node b at coordinates $(\delta, 0)$ and performs 10^7 independent trials. At each trial, two nodes are generated uniformly at random in the circle. At the end of all trials, the desired probability is computed as the fraction of trials that resulted in nodes k and ℓ as in Section 4.

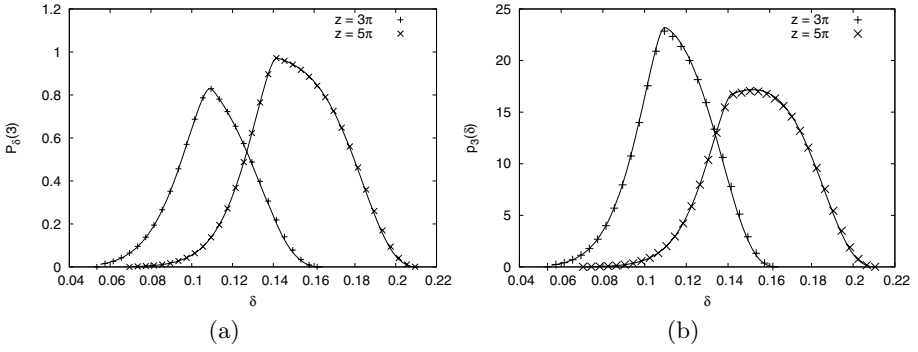


Fig. 6. $P_\delta(3)$ (a) and $p_3(\delta)$ (b). Solid lines give the analytic predictions.

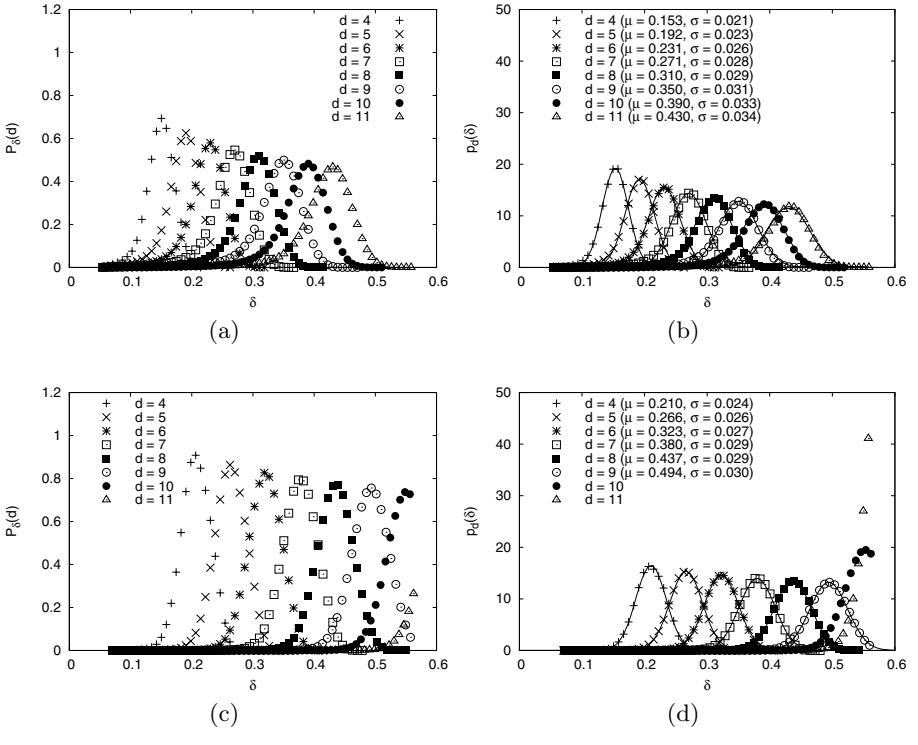


Fig. 7. $P_\delta(d)$ and $p_d(\delta)$ for $d > 3$, with $z = 3\pi$ (a, b) and $z = 5\pi$ (c, d). Solid lines give the Gaussians that best fit some of the $p_d(\delta)$ data, each of mean μ and standard deviation σ as indicated.

The simulation for the determination of n' is conducted for $\delta = 2R$ only, whence $\rho_\delta = 0$. This is the value of δ for which the results from the simulation above for $P_\delta(3)$ and the analytic prediction given by $1 - [1 - P'_\delta(3)]^{n-2}$ differ the most (and significantly; data not shown). Moreover, as we will see shortly, the

value of n' we find using this value of δ is good for all other values as well. The simulation is aimed at finding the value of Q_δ and proceeds in 10^9 independent trials. Each trial fixes node b at $(\delta, 0)$ and places the remaining $n-2$ nodes in the circle uniformly at random. The fraction of trials resulting in no node qualifying as the node k of Section 4 is the value of Q_δ. We set n' to be the $m < n-2$ that minimizes $|Q_\delta - [1 - P'_\delta(3)]^m|$, where $P'_\delta(3)$ refers to the analytic prediction. Our results are $n' = 779$ for $z = 3\pi$, $n' = 780$ for $z = 5\pi$.

Results for $d = 1$ are shown in Figure 3, for $d = 2$ in Figure 4, for $d = 3$ in Figures 5 and 6, and for $d > 3$ in Figure 7. In all figures, both $P_\delta(d)$ and $p_d(\delta)$ are plotted against δ, since it seems better to visualize what happens as one gets progressively farther from node a in Euclidean terms. For this reason, the plots for $P_\delta(d)$ do not constitute a probability distribution for any fixed value of d.

7 Discussion and Conclusion

The results summarized in Figures 3 through 6 reveal excellent agreement between the analytic predictions we derived in Sections 3 through 5 and our simulation data. This holds not only for the simple cases of $d = 1$ and $d = 2$, but also for the considerably more complex cases of $P'_\delta(3)$ and $P_\delta(3)$. The latter, in particular, depends on the empirically determined n'. In this respect, it is clear from Figure 6 that, even though n' could have been calculated for a greater assortment of δ values, doing it exclusively for $\delta = 2R$ seems to have been sufficient.

Figure 7 contemplates some of the $d > 3$ cases, for which we derived no analytic predictions. The values of d that the figure covers in parts (a, b) and (c, d), respectively for $z = 3\pi$ and $z = 5\pi$, are $4, \ldots, 11$. Of these, $d = 11$ for $z = 5\pi$ in part (d) typifies what happens for larger values of d as well (omitted for clarity), viz. probability densities sharply concentrated at the border of the radius-$\sqrt{1/\pi}$ circle centered at node a. Note that the same also occurs for $z = 3\pi$, but owing to the smaller R it only happens for larger values of d (omitted from part (b), again for clarity).

For $4 \leq d \leq 11$ with $z = 3\pi$, and $4 \leq d \leq 9$ with $z = 5\pi$, Figures 7(b) and (d) also display Gaussian approximations of $p_d(\delta)$. Parts (a) and (c) of the figure, in turn, contain the corresponding simulation data only, and we remark that the absence of some approximation computed from the Gaussians of part (b) or (d) is not a matter of difficulty of principle. In fact, the counterpart of Equation (2), obtained also from Bayes' theorem and such that

$$P_\delta(d) = \frac{p_d(\delta)P(d)}{p(\delta)} = \frac{p_d(\delta)P(d)}{\sum_{s=1}^{n-1} p_s(\delta)P(s)}, \tag{12}$$

can in principle be used with either those Gaussians or the concentrated densities in place of $p_s(\delta)$ as appropriate for each s. What prevents this, however, is that we lack a characterization of $P(s)$ that is not based on simulation data only.

Still in regard to Figure 7, one possible interpretation of the good fit by Gaussians of the simulation data for $p_d(\delta)$ comes from resorting to the central limit theorem in its classical form [35]. In order to do this, we view δ as valuing

the random variable representing the average Euclidean distance to node a of all nodes that are d edges apart from a. The emergence of $p_d(\delta)$ as a Gaussian for $d > 3$ (provided d is small enough that the circle's border is not influential) may then indicate that, for each value of d, the Euclidean distances of those nodes to node a are independent, identically distributed random variables. While we know that this does not hold for the smaller values of d as a consequence of the uniformly random positioning of the nodes in the circle (smaller Euclidean distances to a are less likely to occur for the same value of d), it would appear that it begins to hold as d is increased.

To summarize, we have considered a network of sensors placed uniformly at random in a two-dimensional region and, for its representation as a random geometric graph, have studied two distance-related distributions. One of them is the probability distribution of distances between two randomly chosen nodes, conditioned on the Euclidean distance between them. The other is the probability density function associated with the Euclidean distance between two randomly chosen nodes, given the distance between them. We have provided analytical characterizations whenever possible, in the simplest cases as closed-form expressions, and have also validated these predictions through simulations.

While further work related to additional analytical characterizations is worth undertaking, as is the investigation of the three-dimensional case, we find that the most promising tracks for future investigation are those that relate to applications. In Section 1 we illustrated this possibility in the context of sensor localization, for which it seems that understanding the distance-related distributions we have studied has the potential to help in the discovery of better distributed algorithms. Whether there will be success on this front remains to be seen, as well as whether other applications will be found with the potential to benefit from the results we have presented.

Acknowledgments

The authors acknowledge partial support from CNPq, CAPES, and a FAPERJ BBP grant.

References

1. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. Discrete Mathematics 86, 165–177 (1990)
2. Penrose, M.: Random Geometric Graphs. Oxford University Press, Oxford (2003)
3. Erdős, P., Rényi, A.: On the evolution of random graphs. Publications of the Mathematical Institute of the Hungarian Academy of Sciences 5, 17–61 (1960)
4. Newman, M.E.J., Strogatz, S.H., Watts, D.J.: Random graphs with arbitrary degree distributions and their applications. Physical Review E 64, 026118 (2001)
5. Appel, M.J.B., Russo, R.P.: The connectivity of a graph on uniform points in $[0, 1]^d$. Statistics & Probability Letters 60, 351–357 (1996)
6. Appel, M.J.B., Russo, R.P.: The maximum vertex degree of a graph on uniform points in $[0, 1]^d$. Advances in Applied Probability 29, 567–581 (1997)

7. Appel, M.J.B., Russo, R.P.: The minimum vertex degree of a graph on uniform points in $[0,1]^d$. Advances in Applied Probability 29, 582–594 (1997)
8. Meester, R., Roy, R.: Continuum Percolation. Cambridge University Press, Cambridge (1996)
9. Gupta, P., Kumar, P.R.: Critical power for asymptotic connectivity in wireless networks. In: McEneany, W.M., Yin, G., Zhang, Q. (eds.) Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W. H. Fleming, pp. 547–566. Birkhäuser, Boston (1998)
10. McDiarmid, C.: Random channel assignment in the plane. Random Structures and Algorithms 22, 187–212 (2003)
11. Bailey, D.H., Borwein, J.M., Crandall, R.E.: Box integrals. Journal of Computational and Applied Mathematics 206, 196–208 (2007)
12. Bondy, J.A., Murty, U.S.R.: Graph Theory with Applications. North-Holland, New York (1976)
13. Estrin, D., Girod, L., Pottie, G., Srivastava, M.: Instrumenting the world with wireless sensor networks. In: Proceedings of ICASSP 2001, vol. 4, pp. 2033–2036 (2001)
14. Karp, B., Kung, H.T.: GPSR: Greedy perimeter stateless routing for wireless networks. In: Proceedings of MobiCom 2000, pp. 243–254 (2000)
15. Eren, T., Goldenberg, D., Whitley, W., Yang, Y.R., Morse, A.S., Anderson, B., Belhumeur, P.N.: Rigidity, computation, and randomization in network localization. In: Proceedings of INFOCOM 2004, vol. 4, pp. 2673–2684 (2004)
16. Aspnes, J., Eren, T., Goldenberg, D., Morse, A.S., Whiteley, W., Yang, Y.R., Anderson, B.D.O., Belhumeur, P.N.: A theory of network localization. IEEE Transactions on Mobile Computing 5, 1663–1678 (2006)
17. Doherty, L., Ghaoui, L.E., Pister, K.S.J.: Convex position estimation in wireless sensor networks. In: Proceedings of INFOCOM 2001, vol. 3, pp. 1655–1663 (2001)
18. Shang, Y., Ruml, W., Zhang, Y., Fromherz, M.P.J.: Localization from mere connectivity. In: Proceedings of MobiHoc 2003, pp. 201–212 (2003)
19. He, T., Huang, C., Blum, B., Stankovic, J., Abdelzaher, T.: Range-free localization schemes in large scale sensor networks. In: Proceedings of MobiCom 2003, pp. 81–95 (2003)
20. Niculescu, D., Nath, B.: DV based positioning in ad hoc networks. Telecommunication Systems 22, 267–280 (2003)
21. Meertens, L., Fitzpatrick, S.: The distributed construction of a global coordinate system in a network of static computational nodes from inter-node distances. Tech. Rep. KES.U.04.04, Kestrel Institute, Palo Alto, CA (2004)
22. Moore, D., Leonard, J., Rus, D., Teller, S.: Robust distributed network localization with noisy range measurements. In: Proceedings of SenSys 2004, pp. 50–61 (2004)
23. Hu, L., Evans, D.: Localization for mobile sensor networks. In: Proceedings of MobiCom 2004, pp. 45–57 (2004)
24. Ji, X., Zha, H.: Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling. In: Proceedings of INFOCOM 2004, vol. 4, pp. 2652–2661 (2004)
25. Bulusu, N., Heidemann, J., Estrin, D.: GPS-less low-cost outdoor localization for very small devices. IEEE Personal Communications 7, 28–34 (2000)
26. Weisstein, E.W.: CRC Concise Encyclopedia of Mathematics, 2nd edn. Chapman & Hall/CRC, Boca Raton (2002)
27. Bahl, P., Padmanabhan, V.: RADAR: An in-building RF-based user location and tracking system. In: Proceedings of INFOCOM 2000, vol. 2, pp. 775–784 (2000)

28. Hightower, J., Want, R., Borriello, G.: SpotON: An indoor 3D location sensing technology based on RF signal strength. Tech. Rep. #2000-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA (2000)
29. Priyantha, N., Chakraborty, A., Balakrishnan, H.: The Cricket location-support system. In: Proceedings of MobiCom 2000, pp. 32–43 (2000)
30. Girod, L., Estrin, D.: Robust range estimation using acoustic and multimodal sensing. In: Proceedings of IROS 2001, vol. 3, pp. 1312–1320 (2001)
31. Varga, A.: The OMNeT++ discrete event simulation system. In: Proceedings of ESM 2001 (2001)
32. Niculescu, D., Nath, B.: Ad hoc positioning system (APS) using AoA. In: Proceedings of INFOCOM 2003, vol. 3, pp. 1734–1743 (2003)
33. Nagpal, R.: Organizing a global coordinate system from local information on an amorphous computer. A.I. Memo No. 1666, MIT A.I. Laboratory, Cambridge, MA (1999)
34. Bettstetter, C., Eberspächer, J.: Hop distances in homogeneous ad hoc networks. In: Proceedings of VTC 2003, vol. 4, pp. 2286–2290 (2003)
35. Trivedi, K.S.: Probability and Statistics with Reliability, Queuing and Computer Science Applications, 2nd edn. John Wiley & Sons, Chichester (2002)
36. Fewell, M.P.: Area of common overlap of three circles. Tech. Note DSTO-TN-0722, Defence Science and Technology Organisation, Edinburgh, Australia (2006)
37. Römer, K., Mattern, F.: The design space of wireless sensor networks. IEEE Wireless Communications 11, 54–61 (2004)
38. Dall, J., Christensen, M.: Random geometric graphs. Physical Review E 66, 16121 (2002)
39. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge (2001)

An Analytic Model for Optimistic STM with Lazy Locking

Armin Heindl[1] and Gilles Pokam[2]

[1] Universität Erlangen-Nürnberg
Computer Networks and Communication Systems
Erlangen, Germany
[2] Intel Corporation
Microprocessor Technology Lab
Santa Clara, CA, USA

Abstract. We extend an existing analytic framework for modeling software transactional memory (STM) to an optimistic STM variant in which write locks are acquired lazily. Lazy locking requires a different calculation of the transition probabilities of the underlying discrete-time Markov chain (DTMC).

Based on few relevant input parameters, like the number of concurrent transactions, the transaction lengths, the share of writing operations and the number of accessible transactional data ojects, a fixed-point iteration over closed-form analytic expressions delivers key STM performance measures, e.g., the mean number of transaction restarts and the mean number of processed steps of a transaction.

In particular, the analytic model helps to predict STM performance trends as the number of cores on multi-processors increases, but other performance trends provide additional insight into system behavior.

1 Introduction

As a parallel programming model for shared-memory chip multi-processors (CMPs), the concept of transactional memory (TM, [1,2]) has received a lot of attention as an alternative to synchronizing parallel applications by means of classical locking mechanims to coordinate access to shared data.

Writing parallel multi-threaded programs based on classical locking is a challenging task. Fine-grained locking may become very complex enhancing the risk of deadlocks or otherwise incorrect program behavior. Coarse-grained locking may result in inefficient code. TM tries to overcome these drawbacks by providing primitives to the programmer to label critical code with external memory accesses, so-called transactions, and resolves resulting conflicts between concurrent transactions at runtime. Outside of the TM system, transactions (including several read and write operations to transactional memory) appear to be executed atomically, while the TM system ensures a consistent view of the concurrent transactions on the transactional data. Non-conflicting transactions may execute unaffectedly until they successfully finish (i.e., commit). Conflicting data accesses are detected and resolved in different ways ranging from delaying to aborting and restarting one or more transactions.

Many different mechanisms for meta-data organization, concurrency control and contention management have been proposed and can be combined to various different

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 339–353, 2009.

TM systems. Some variants have been implemented by the research community in order to better understand TM behavior – either in software [3,4,5], in hardware [1,6,7,8,9] or in a hybrid way [10]. These implementations are very important triggers for pushing forward TM advancement. We believe, however, that they should be complemented with other techniques that facilitate the decision making in the multitude of TM design choices [2], especially for the more flexible software TM (STM).

In [11], we first proposed to model STM performance on the basis of a discrete-time Markov chain (DTMC), whose current states encode the progress of a representative transaction. As this so-called tagged transaction executes in the system, it is influenced by other concurrent transactions via aggregate parameters, which characterize the system state in terms of meta-data like read and write set sizes. The models in [11,12] have been developed for STM variants, where write locks are acquired eagerly, i.e., write locks are acquired at the time the transaction accesses the data (and released at commit or abort time). Here, we consider another popular STM variant, namely optimistic STM with write buffering, where write locks are acquired lazily, i.e., only for the duration of the final commit step. The shorter lock holding times enable a higher degree of concurrency for speculative transactions at the risk of detecting conflicts later than with early locking.

The analytic model for optimistic STM with write buffering and lazy locking is elaborated within the framework proposed in [12]. Starting from a DTMC with identical structure, other closed-form expressions must be derived for the transition probabilities in order to reflect the fact that elements in the write set of a transaction no longer imply write locks on the respective transactional data. The resulting fixed-point iteration over a set of algebraic equations efficiently delivers the same set of STM performance characteristics, like the mean number of restarts of a transaction or the mean total number of steps of a possibly restarted transaction.

Analytical models for TM algorithms were also proposed in [13,14]. However, the execution model in [13] does not consider a transactional memory execution, but rather the inter-dependencies between sets of potential parallel tasks. As a result, the approach cannot be used to provide insight into the dynamics of TM algorithms, as we do in this paper. In [14], the authors study the performance of an STM system in the context of false sharing. They analyze the tensions between the size of a transaction and the likelihood of conflict resulting in closed-form expressions. Our approach is based on a different methodology leading to a rather flexible framework, which provides a wide range of performance measures. This framework may be adapted to other STM variants beyond the ones studied so far.

The rest of the paper is organized as follows. Section 2 describes the base STM system, for which we construct the performance model in Section 3. Therein, we also develop the closed-form expressions for the evaluation of the model. After a validation of the model against discrete-event simulation, we present the trend behavior of critical STM performance measures in Section 4. Finally, we conclude in Section 5.

2 Optimistic STM with Write Buffering and Lazy Locking

Diverse names can be found in the literature for the optimistic STM variant studied in this paper: write buffering is sometimes referred to as deferred update, write-back or

lazy version management. Lazy locking is also called commit-time locking (as opposed to eager locking alias encounter-time or open/acquire-time locking).

The STM variant discussed in this paper is obstruction-free and object based with per-object meta-data [15]. Since the terms *write buffering* and *lazy locking* in the context of optimistic STM still leave several design decisions open, e.g., with respect to concurrency control, conflict detection and resolution and versioning rules, we clarify in this section upon which operational rules the DTMC-based model of the next section will be constructed.

In STM, a transaction is a sequence of read and write operations that appears indivisible and instantaneous to the outside world. Other transactions (or threads) notice that *all* operations of a transaction have been executed or none (atomicity). A transaction always leaves the transactional data in a consistent state (consistency), irrespective of the number of concurrent transactions (isolation).

By means of mutually exclusive read and write locking of the respective data objects, potentially conflicting data accesses could be resolved as early as possible, namely at encounter time (as in pessimistic STM). However, optimistic STM allows write accesses to transactional data even if other transactions are currently reading the data object. Under certain circumstances, such speculative readers in write-after-read (WAR) situations may finish successfully after all, thus reducing the number of restarts and increasing the throughput of the STM system. On the other hand, if restarts are only delayed (as compared to pessimistic STM), more computation time is wasted.

Optimistic STM operates without read locks, while write locks may in principle be acquired any time between the initial access to the data (or its meta-data descriptor) and the release of all locks the transaction holds (at commit and abort times). With lazy locking, the acquisition of write locks is deferred to the beginning of the commit operation. Lazy locking mandates write buffering[1] to temporarily hide the impact of write operations from other transactions. Write buffering means that transactional write operations are not performed on the global data, but in thread-local buffers. Throughout their lifetime, transactions manipulate these local copies of the transactional data until the changes are made visible by a successful commit operation.

Essentially, the local write buffers constitute the write set of a transaction. A transaction also keeps track of the transactional data it has read since it has (re)started. The corresponding set is called read set. By means of its read set and a proper versioning scheme, the transaction validates that its view on the transactional data is consistent

In this paper, we assume that it is not visible to a transaction which other transactions are currently reading or writing on transactional data in its own read or write set. As a consequence, we only consider passive aborts, i.e., a transaction can only abort itself upon a detected conflict, but not other conflicting transactions. Neither can locks be stolen from another transaction. It suffices that the meta-data for transactional data indicates if the data is write-locked or not. The data structures required for corresponding meta-data organization – both for write locking and read/write set organization – is beyond the scope of this paper and is described elsewhere (see e.g., [15]).

Thus, in optimistic STM with write buffering and lazy locking, a read or write request to some data object is granted to a transaction, unless this data object is write-locked by

[1] Of course, write buffering may also be combined with eager locking.

another transaction. Note that write-locking only occurs during the commit phase. Only the write locks are mutually exclusive. On the contrary, read and write sets of concurrent transactions may arbitrarily overlap (including WAR, WAW, RAW and RAR situations). If a transaction writes to a data object (i.e., it actually stores the data value to be written in a local buffer) which it has read before, the corresponding element is moved from the read set to the write set.

The overlap of read and write sets among concurrent transactions requires a validation procedure to detect states of data inconsistency, i.e., to detect that a reading transaction is working with outdated transactional data (due to successfully performed writes by other transactions). We assume that such a validation check is based on version numbers of the transactional data and is performed right at the end of every read request and at the final commit operation.

Versioning essentially means to associate a global counter (visible to any transaction) with *each* transactional data object. Write operations eventually increment these counters to indicate that the transactional data has been modified. When a transaction first successfully reads transactional data, it records the value of its counter (i.e., the current global version number of the transactional data) in a newly created read set entry. In every validation procedure, the transaction compares all locally stored version numbers with the respective current global version numbers. If any global version number has been incremented by another transaction in the meantime (i.e., any locally stored version number is smaller than the corresponding global version number), the validation check fails and the validating transaction aborts and restarts. Otherwise the transaction may continue with the next operation.

In the commit step, a transaction

1. must successfully acquire all write locks for the data in its write set,
2. must successfully validate its read set,
3. then increments the global version numbers of the write-locked data
4. copies the values in the local write buffers to the global memory locations,
5. and releases all write locks.

If the commit operation fails (in the first two steps), the transaction releases all obtained write locks and aborts. An abort (also on other occasions) always implies that current read and write sets are dissolved and the transaction restarts. Due to the local write buffers, an abort generally leaves the global memory unmanipulated. The global memory locations are only modified (i.e., the local values are copied to the global memory locations), if a transaction successfully finishes making the local changes permanent and visible to other transactions. Only then are the version numbers incremented.

In any WAR (write-after-read) or RAW (read-after-write) situations, the reading transaction will not have to abort due to the respective data, as long as the writing transaction does not successfully commit before the reading transaction does. In fact, if the reading transaction also issues a write request on the respective data in a subsequent operation, the "reading" transaction may even successfully commit after the writing transaction. In this case, if both attempt to commit at the same time, their write-lock acquisitions may conflict.

While concurrent transactions may operate on different values for the transactional data (due to local copies), (incremental) validation checks guarantee that each

transaction has a consistent view of the transactional data. For instance, data that is only read will always have the same value for this transaction between (re)start and commit; otherwise the transaction will be aborted.

3 The DTMC-Based Model

Instead of modeling explicitly all transactions in their concurrent behavior, our model characterizes the representative behavior of a single so-called tagged transaction. The impact of the other transactions will be captured by appropriately computing the parameters of the single-transaction model. To some extent, this approach assumes that all transactions have a similar probabilistic behavior.

Since we want to study performance measures independently of the specific timing between data accesses, we consider the behavior of the tagged transaction at the instants of read/write requests. As in [12], we propose to model this behavior as an (absorbing) embedded Markov chain, whose states are enumerated according to the current number of read/write operations that have been successfully performed, i.e., the current progress of the transaction. This includes repeated accesses to the same data. The transition probabilities of the absorbing discrete-time Markov chain (DTMC) will mainly depend on how many write locks on the transactional data are currently held by the transactions, but also on the sizes of their write and read sets, especially before the commit operation. This information can in turn be inferred from the DTMC. As a consequence, a fixed-point iteration scheme arises.
The models is determined by only four key input parameters L, N, k and l_w:

- Integer L denotes the number of transactional data items in the system, i.e., the amount of data accessible to all transactions.
- Integer N denotes the number of concurrent transactions competing for the transactional data. We assume that there are always N active transactions on dedicated cores, i.e., N is constant[2].
- The tagged transaction has to perform k subsequent read and write operations successfully in order to finish (where k includes repeated reads/writes to same data).
- We do not assume any particular ordering of the read and write operations in a transaction. Instead, any request is issued as a write access with probability l_w (and as a read access with probability $l_r = 1 - l_w$). All transactional data objects are equally popular, i.e., have the same probability of being accessed (i.i.d. data accesses with uniform distribution on L).

3.1 Transaction Behavior

The execution of a tagged transaction $T^{(j)}$ $(1 \leq j \leq N)$ is observed at the epochs when a read or write request is issued. (In the following, we omit the superscript j to simplify notation.) With each successful read or write operation on transactional data, transaction T advances to the next epoch, i.e., we define state i of transaction T as the state in which T has performed a sequence of i successful operations.

[2] Otherwise, we might work with an effective number N_{eff} of concurrent transactions.

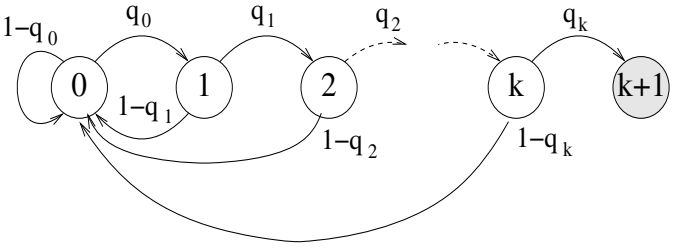


Fig. 1. Absorbing DTMC representing the execution of a single transaction

Depending on the current state i of the tagged transaction and the behavior of the other transactions, the current read/write request will be successful with (non-zero) probability q_i, i.e., with this non-conflict probability the tagged transaction moves from state i to state $i+1$ ($0 \leq i < k$). With the complementary probability $1 - q_i$, a conflict occurs and the transaction must restart in state 0, i.e., resumes execution from state 0.

Figure 1 depicts this behavior in form of an absorbing discrete-time Markov chain (DTMC). Here, we assume that when the transaction is aborted, it restarts immediately. At stage k, a commit operation (which includes the write lock acquisition and validation of the read set) succeeds with probability q_k and fails with $1 - q_k$, possibly depending on factors like the sizes of the read and write sets of the tagged transaction and the writing behavior of the other transactions. The transaction completes when it reaches the absorbing state $k+1$.

In state i ($0 \leq i \leq k$), the combined size of read and write sets of a transaction may be smaller than i due to repeated read and write accesses to the same data. The number of accesses to distinct data corresponds to the sizes of read and write sets. Since a transaction accesses data equiprobabilistically, we can compute the mean combined size of read and write sets in state i as

$$n_q^{(i)} = L\left(1 - (1 - \frac{1}{L})^i\right) \tag{1}$$

according to the general birthday problem [16]. An average number of $n_q^{(i)}$ data objects are actually touched by i independent data accesses to L data objects. On average, the size of the write set is approximately $n_{qw}^{(i)} = l_w n_q^{(i)}$ and that of the read set $n_{qr}^{(i)} = l_r n_q^{(i)}$. We will use this and similar information about the non-tagged transactions below in order to establish expressions for the non-conflict probabilities q_i.

3.2 Average Sizes of Read and Write Sets Held by an Arbitrary Transaction

Assuming probabilistically identical behavior[3], we may interpret the DTMC of Figure 1 as the representative behavior of an *arbitrary* (not only the tagged) transaction. With the DTMC fully specified, important performance measures, like the mean number $E[S]$ of steps of a transaction before absorption, can be computed as outlined below. $E[S]$

[3] Otherwise, we exploit some experimental proportionality between the average current progress of a transaction and k, the given number of operations, similar to a discussion in [11].

counts all requests, including those, which might have to be repeated due to a restart of the transaction, plus the final commit step, and thus corresponds to the number of steps of the transient DTMC until absorption. Let us also introduce $E[I]$ as the average current progress of an arbitrary transaction, i.e., the average of the state numbers, in which it may reside at an arbitrary instant of time before absorption.

The absorbing DTMC of Figure 1 consists of $k+2$ states, where state $k+1$ is the absorbing state. We denote by p_i $(0 \leq i \leq k)$ the probability that – at an arbitrary instant of time before absorption – the DTMC is in state i, i.e., the probability that the transaction has successfully processed the first i read and write operations at an arbitrary instant before its completion.

For an absorbing DTMC, p_i may be computed as the ratio of the mean number of visits to state i, $E[S_i]$, over the mean number of steps of the transient DTMC until absorption, $E[S]$. $E[S]$ is computed as [17]

$$E[S] = \mathbf{v_0} \left(\mathbf{I} - \mathbf{P}\right)^{-1} \mathbf{e} \quad , \tag{2}$$

where row vector $\mathbf{v_0}$ is the initial probability vector of the transient DTMC with (sub-stochastic) transition probability matrix $\mathbf{P}$. Matrix $\mathbf{I}$ is the identity matrix of appropriate dimension and column vector $\mathbf{e}$ a corresponding vector of ones. For the DTMC of Figure 1, matrices and vectors are of dimension $k+1$ and more specifically

$$\mathbf{v_0} = \begin{bmatrix} 1\, 0 \ldots 0 \end{bmatrix} \quad , \quad \mathbf{P} = \begin{bmatrix} 1-q_0 & q_0 & 0 & \cdots & 0 \\ 1-q_1 & 0 & q_1 & 0 & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 1-q_{k-1} & 0 & \cdots & 0 & q_{k-1} \\ 1-q_k & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Evaluating equation (2) and using the fact that – with the specific initial probability vector $\mathbf{v_0}$ – $E[S_i]$ simply equals the ith component of the first row of the fundamental matrix $(\mathbf{I} - \mathbf{P})^{-1}$), we get

$$E[S] = \Sigma_{i=0}^{k} \frac{1}{\Pi_{j=i}^{k} q_j} \quad \text{and} \quad E[S_i] = \frac{1}{\Pi_{j=i}^{k} q_j} \, . \tag{3}$$

If all non-conflict probabilities equal 1, $E[S] = k+1$ and $E[S_i] = 1$ as expected.

Hence, the probabilities $p_i = \frac{E[S_i]}{E[S]}$ for $i = 0, \ldots, k$ are

$$p_i = \frac{\Pi_{j=0}^{i-1} q_j}{\Sigma_{i=0}^{k} \Pi_{j=0}^{i-1} q_j} \quad . \tag{4}$$

We compute the average current progress of a transaction, denoted by $E[I]$, as

$$E[I] = \sum_{i=0}^{k} i \; p_i = \frac{\Sigma_{i=0}^{k} i \Pi_{j=0}^{i-1} q_j}{\Sigma_{i=0}^{k} \Pi_{j=0}^{i-1} q_j} \quad . \tag{5}$$

3.3 Non-conflict Probabilities for Optimistic STM with Lazy Locking

In this section, we determine the non-conflict probabilities in response to read/write requests, i.e., probabilities that, given a transaction is in state i ($0 \leq i < k$), it does not encounter a conflict upon its $(i+1)$th request and continues executing from state $i+1$ in the next step.

In case a transactional read operation is issued in the considered STM variant, a transaction T moves from state i to the next state $i+1$ in one step, if and only if both conditions below hold:

data unlocked: The requested data object is not write-locked by another transaction and

validation successful: The additional validation of the read set (via the version numbers) does not fail.

For successful progress with a transactional write operation, only the first condition must be fulfilled.

Now, given that a request occurs, it may be either a write request with probability l_w or a read request with probability $l_r = 1 - l_w$. We can then write q_i as follows

$$q_i = l_w \; P_i\{\text{data unlocked}\} + l_r \; P_i\{\text{data unlocked}\} \cdot P_i\{\text{validation successful}\} \;, \quad (6)$$

where $0 \leq i < k$ and subscript i denotes that the tagged transaction is in state i.

To compute $P_i\{\text{data unlocked}\}$, we recall that in optimistic STM with lazy locking, a transaction only holds write locks for the duration of its commit operation. At an arbitrary time, $(N-1)p_k$ transactions are in the commit phase (on average); each of them has issued $l_w k$ successful write requests. Since up to the commit phase, write requests (and write sets) in different transactions may refer to the same data, the average number of distinct data to be write-locked is again computed in the setting of the general birthday problem, namely by $L\left(1-(1-\frac{1}{L})^{(N-1)p_k l_w k}\right)$. Here, we neglect the fact that actually fewer write locks my be held, since overlapping write sets of committing transactions will lead to conflicts and thus aborts with immediate release of already acquired write locks. The possibility of such conflicts is, however, taken into account in the probability for successful commit operations (see q_k below).

$P_i\{\text{data unlocked}\}$ is then obtained as the ratio of the mean numbers of unlocked data to all accessible data, i.e.,

$$P_i\{\text{data unlocked}\} = \frac{L - L\left(1-(1-\frac{1}{L})^{(N-1)p_k l_w k}\right)}{L} = (1-\frac{1}{L})^{(N-1)p_k l_w k} \;. \quad (7)$$

Since – with write buffering – each transaction actually only writes on the globally visible data with the successful commit operation, the effective write rate can be approximated with $\frac{1}{E[S]}$. If we know the probability $p_{c,0}^{(i)}$ that the commit operation of an arbitrary transaction does not affect the read set of the tagged transaction in state i, we can approximate $P_i\{\text{validation successful}\}$ by

$$P_i\{\text{validation successful}\} = \left(1 - \frac{1-p_{c,0}^{(i)}}{E[S]}\right)^{N-1} \quad . \quad (8)$$

Here, $\frac{1-p_{c,0}^{(i)}}{E[S]}$ is the probability that an arbitrary other transaction would cause the tagged transaction to fail in step i. With the product of the complementary probabilities, we require that none of the $N-1$ other transactions interferes with the tagged transaction.

Let us now determine the probability $p_{c,0}^{(i)}$ as the ratio of successful commit operations of a committing transaction that do not write to the read set of the tagged transaction in state i to all commit operations. Potentially, any set of $n_{qw}^{(k)}$ data objects to be written out of $L - n_{qr}^{(k)}$ data objects, to which the committing transaction can write, may result in a successful commit operation, i.e., we have $\binom{L-n_{qr}^{(k)}}{n_{qw}^{(k)}}$. Here, we subtract the size of the read set from L, since read set and write set of the committing transaction necessarily cover distinct data objects.

For the numerator, the number of positive events is the number of combinations in which $n_{qw}^{(k)}$ data objects to be written can be drawn (without putting back) from $L - L\left(1-(1-\frac{1}{L})^{l_r k + l_r i + (N-2)p_k l_w k}\right)$, i.e., we have $\binom{L-L\left(1-(1-\frac{1}{L})^{(N-2)p_k l_w k + l_r i + l_r k}\right)}{n_{qw}^{(k)}}$. Here, we subtract not only the size of the read set $n_{qr}^{(k)}$ of the committing transaction, but also the size of the read set $n_{qr}^{(i)}$ of the tagged transaction as well as all data objects which are currently write-locked by other committing transactions (on average $n_{qw}^{(k)}$ for a single transaction, of which there are $(N-2)p_k$ in the commit operation). Accessing these write-locked data objects would lead to an unsuccessful commit operation. Instead of subtracting $n_{qr}^{(k)} + n_{qr}^{(i)} + (N-2)p_k n_{qw}^{(k)}$ above, we subtract $L\left(1-(1-\frac{1}{L})^{l_r k + l_r i + (N-2)p_k l_w k}\right)$ in order to eliminate write/read requests to non-distinct data of the tagged and the other committing transactions. Once again, the solution of the general birthday problem delivers this expression. Note that the tagged transaction is in state i, while the others are committing in state k and recall that with lazy locking, read and write sets of the involved transactions are rather decoupled so that both the read sets of the tagged transaction and of the considered committing transaction and the write sets of the other committing transactions may all overlap. Therefore we subtract only the number of distinct data objects.

Rounding the parameters to reasonable integer values so that the corresponding binomial coefficients can be computed yields

$$p_{c,0}^{(i)} = \frac{\binom{\lceil L-L\left(1-(1-\frac{1}{L})^{(N-2)p_k l_w k + l_r i + l_r k}\right) \rceil}{\lfloor n_{qw}^{(k)} \rfloor}}{\binom{\lceil L - n_{qr}^{(k)} \rceil}{\lfloor n_{qw}^{(k)} \rfloor}} = \frac{\binom{\lceil L(1-\frac{1}{L})^{(N-2)p_k l_w k + l_r i + l_r k} \rceil}{\lfloor n_{qw}^{(k)} \rfloor}}{\binom{\lceil L - n_{qr}^{(k)} \rceil}{\lfloor n_{qw}^{(k)} \rfloor}} \quad (9)$$

Combining (7) and (8), we write (6) as follows:

$$q_i = l_w\,(1-\frac{1}{L})^{(N-1)p_k l_w k} \;+\; l_r\,(1-\frac{1}{L})^{(N-1)p_k l_w k}\left(1-\frac{1-p_{c,0}^{(i)}}{E[S]}\right)^{N-1} \quad (10)$$

With similar arguments as above, we compute probability q_k that the commit operation is successful. Before validation is performed (in analogy to (8) for $i = k$), the write locks for all data objects in the write set of the tagged transition have to be acquired.

Thus, we obtain

$$
\begin{aligned}
q_k &= P\{\text{successful commit}\} \qquad (11)\\
&= \frac{\binom{\lceil L - L\left(1-(1-\frac{1}{L})^{(N-1)p_k l_w k + l_r k}\right) \rceil}{\lfloor n_{qw}^{(k)} \rfloor}}{\binom{\lceil L - n_{qr}^{(k)} \rceil}{\lfloor n_{qw}^{(k)} \rfloor}} \cdot \left(1 - \frac{1 - p_{c,0}^{(k)}}{E[S]}\right)^{N-1}\\
&= \frac{\binom{\lceil L(1-\frac{1}{L})^{(N-1)p_k l_w k + l_r k} \rceil}{\lfloor n_{qw}^{(k)} \rfloor}}{\binom{\lceil L - n_{qr}^{(k)} \rceil}{\lfloor n_{qw}^{(k)} \rfloor}} \cdot \left(1 - \frac{1 - p_{c,0}^{(k)}}{E[S]}\right)^{N-1} .
\end{aligned}
$$

The first fraction in this expression follows in analogy to $p_{c,0}^{(i)}$ itself. However, now the perspective of the tagged transaction is assumed: instead of excluding the read-set items of another transaction, we require that none of the possibly overlapping write sets of all other $N-1$ transactions (to be turned into write locks) are accessed (while the tagged transaction cannot write on its own read set a priori).

3.4 Algorithm and Performance Measures

Solving the DTMC-based STM model means iterating over the following equations until a sufficient precision is reached:

(3)	for	$E[S]$	(with $q_i = 1.0$ initially)
(4)	for	p_k	
(9)	for	$p_{c,0}^{(i)}$	with $n_{qw}^{(k)} = l_w L \left(1 - (1 - \frac{1}{L})^k\right)$
(10)	for	q_i	$(0 \le i < k)$
(11)	for	q_k	with $n_{qr}^{(k)} = l_r L \left(1 - (1 - \frac{1}{L})^k\right)$

On the right-hand side, we indicate definitions that are used in the equations listed on the left-hand side. In all our experiments with reasonable parameters settings, we have always encountered convergence of this scheme.

Several STM performance measures may be obtained from the DTMC-based models. For instance, the values of the non-conflict probabilities or – as a global characteristic – the mean number of data objects referenced in read and/or write sets at an arbitrary time may be of interest. In the next section, we focus on the following key STM performance measures of an arbitrary transaction:

- mean number of restarts: $E[R] = E[S_0] - 1$ (see (3) for $E[S_0]$)
- mean number of steps of a transaction (counting lock requests and the final commit step): $E[S] = \Sigma_{i=0}^{k} \frac{1}{\Pi_{j=i}^{k} q_j}$
- mean sizes of read and write sets of a transaction: $E[Q] = L\left(1 - (1 - \frac{1}{L})^{E[I]}\right)$ (see (5) for $E[I]$)

The parameters $E[R]$ and $E[S]$ should be interpreted together to assess the quality of the concurrency control scheme. Obviously, the smaller the mean number of restarts, the better the performance of the STM system. A smaller value of $E[S]$ (for similar values of $E[R]$ and fixed k) indicates that on the average the restarts occurred earlier in the sequence of read and write operations. Then, less work is lost per restart. Generally, the mean number of transaction steps summarizes the total cost of read/write barrier operations inside a transaction.

The average number $E[Q]$ of read/write set sizes of a transaction may serve as a measure of the total cost of maintaining STM-related meta-data information in the system (when multiplied with the lifetime of the transaction $E[S]$).

4 Numerical Results

In this section, we apply the DTMC-based model proposed in this paper to study the behavior of a transactional memory system. We first validate the analytic model with data from a discrete-event simulation, which will show that the DTMC-based model captures typical trends observed in optimistic STM with write buffering and lazy locking. Further experiments with the analytic model demonstrate its application in a sensitivity analysis of STM performance.

Especially for larger numbers of N or k and/or often small values for performance measures (like the mean number of restarts), a discrete-event simulation requires relatively long run times for statistically significant results, whereas the numeric evaluation of the analytic iteration scheme of the previous section produces results in Maple [18] quasi immediately after a few iterations. For larger values of N or k, the number of iterations may increase beyond 20 for convergence of non-conflict probabilities q_i within an absolute deviation of 10^{-4}.

As opposed to the DTMC-based model, the simulation model implements N transactions explicitly as concurrent processes, of which each one initially selects a sequence of k read and write accesses according to the probability l_w. This sequence is maintained throughout the lifetime of the (simulated) transaction, i.e., after an abort, the transaction retries the identical sequence of operations (unlike the analytical model). Moreover, the simulation selects exponentially distributed delays for read and write operations. Regarding concurrency control and contention management, the simulation model implements the logic described in Section 2. More details about the settings for the simulation model can be found in [19]. The simulation experiments have been performed in the tool AnyLogic [20] with a confidence level of 95% and a relative error of 5% (over all shown performance results).

Figure 2 shows the behavior of the performance measures $E[R]$ and $E[Q]$ as the probability l_w increases. The total number of transactional data L, the number of threads N and the static length of transactions k were set to $L = 1$ mio, $N = 16$, $k = 100$, respectively. These values fall in the ranges of typical benchmark applications [21]. We do not show curves for the mean number of steps of a transaction $E[S]$, since their shapes (with different scaling) are very similar to the curves for $E[R]$ (left-hand side of Figure 2). The solid and dashed lines correspond to the analytical and simulation results, respectively. Starting from $l_w = 0$ with no conflicts, the mean number of restarts $E[R]$

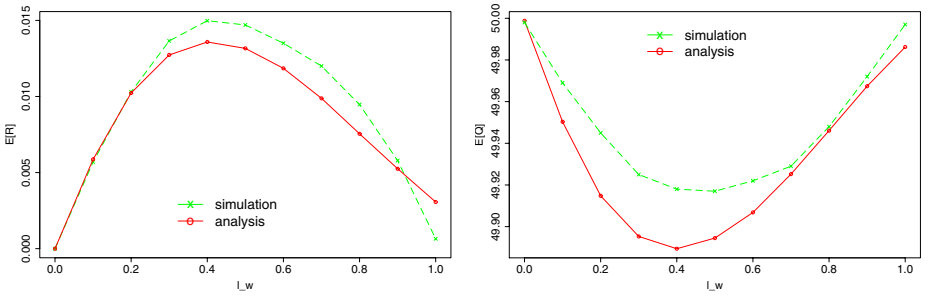


Fig. 2. Mean number of restarts $E[R]$ per transaction (left) and mean size of read/write sets $E[Q]$ per transaction (right), each vs. write probability $l_w \in [0, 1]$ (for $L = 1$ mio, $N = 16$, $k = 100$)

per transactions increase with an increasing share of write operations (up to $l_w = 0.4$). Conflicts are mainly caused by failed validations after reads or in commit operations. With fewer and fewer read operations, $E[R]$ decreases again, but does not reach 0 again at $l_w = 1$ due to the possibility of collisions at write-lock acquisition for the commit. While simulation and analytical results for $E[R]$ are in good agreement in the range $l_w \in [0.0, 0.3]$, the analytical model underestimates the number of restarts for $l_w \in [0.4, 0.8]$ by up to 15% in terms of relative errors. With almost only writing operations, i.e., l_w near 1, the relationship of the curves is inverted.

On the right-hand side of Figure 2, the mean combined size of read and write sets per transaction, $E[Q]$, behaves invertedly to $E[R]$. A minimum of $E[Q]$ is shaped near the center range of l_w, where $E[R]$ assumes the maximum. The averaged sizes of read and write sets of a transaction will be smaller, the more often this transaction restarts, since it spends relatively more time in states with smaller read/write set sizes. At first sight, the deviation between simulation and analysis appear more pronounced for $E[Q]$ (than for $E[R]$), but relative errors are much smaller (below 1%). Suprisingly, the simulation curve for $E[Q]$ is approximated better for large values of l_w than for small values of l_w.

In the light of the mentioned fundamental differences between simulation model and DTMC-based model, the analytical results prove sufficiently accurate. Still, we point out that the motivation of this analytical approach does not lie in producing accurate numbers for specific existing benchmarks, but in efficiently studying trends and trade-offs for projected system parameter values (not yet realizable in current implementations). In the following, we illustrate the intended use of the DTMC-based model.

In the remaining experiments, we fix the share of write operations to $l_w = 0.3$, since fewer write than read operations (often much less) are most common in applications (see also [21]). For different values of k, Figure 3 shows by which percentages $E[R]$ (left) and $E[S]$ (right) are increased (relative to the result for the previous value of N) when the number of threads/processors N is doubled. Here, the number of transactional data is fixed at $L = 1$ million. For the incremental increases, results for $E[R]$ are provided for values $N = 32, 64, 128, 256, 512$ (with initial reference values at $N = 16$), while for $E[R]$ the incremental increases are already computed for $N = 16$ with respect to the number of transaction steps $k + 1$ in the conflict-free case. For $N = 16$, the values for the mean number of restarts are $E[R] = 1.10 \times 10^{-3}, 3.60 \times 10^{-3}, 7.51 \times 10^{-3}, 12.73 \times 10^{-3}, 19.36 \times 10^{-3}$ for $k = 25, 50, 75, 100, 125$, respectively.

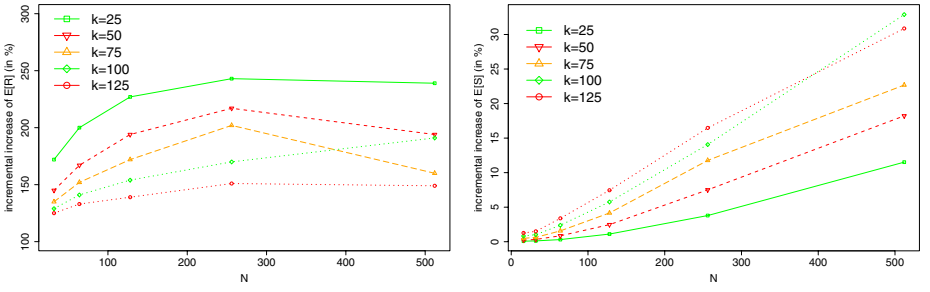


Fig. 3. Incremental increase in mean number of restarts $E[R]$ per transaction (left) and incremental increase of mean number of steps $E[S]$ per transaction (right), each in per cent, whenever N is doubled ($N = 16, 32, 64, 128, 256, 512$) for $L = 1$ mio and different values of k

From Figure 3, we see that – as N doubles, i.e., is increased by 100 % – $E[R]$ increases dramatically (by up to 250 % per step, depending on the specific N and k), while $E[S]$ grows moderately in comparison. The largest jump of 33% for $E[S]$ occurs for $k = 100$, when N is doubled from $N = 128$ to $N = 256$. Especially for lower values of N, introducing more parallelism seems to result in relatively more restarts earlier in the lifetime of transactions so that their average total lifetime is not affected too much. Interestingly, the incremental increases for $E[R]$ tend to be more pronounced for smaller values of k, while for $E[S]$ the incremental increases are larger for larger values of k. In other words, shorter transactions suffer more in terms of number of restarts, while longer transactions suffer more in terms of total number of steps (when N is doubled). For the given value of L and at least for longer transactions ($k = 100$ or 125), the increase of $E[S]$ by around 30 % in the step from $N = 256$ to $N = 512$ already indicates scalability problems of STM with high parallelism. If the processing power is doubled (increased by 100 %), an application will not run twice as fast, but only around 70 % faster. Of course, more specific data access patterns than the assumed i.i.d. read/write accesses on L may adversely affect these numbers. Also, the fundamental trends in Figure 3 only consider performance loss due to conflicts and neglect operational overhead due to meta-data organization.

In the given setting, optimistic STM with write buffering and lazy locking – though better than most other variants – still peforms poorly for $N = 512$: the mean number of restarts per transaction $E[R]$ range from 0.34 (for $k = 25$) to 1.51 (for $k = 125$), while e.g. a transaction with $k = 100$ operations requires 169 steps on average (also see curve for $L = 1$ million in Figure 4). For fixed N, performance decreases faster with increasing k, when N is smaller. For instance, doubling k for $N = 16$ increases $E[R]$ by a factor of around 3.5, while for $N = 512$ this factor is around 2.

Finally, we also show how strongly STM performance depends on L. For different values of L (between 100,000 and 100 millions), Figure 4 plots the increase of the mean number of steps per transaction ($E[S]$ in absolute numbers) as N increases. Naturally, fewer transactional data – under otherwise unchanged conditions – lead to more conflicts and thus longer transaction lifetimes. Initially, i.e., with one processor ($N = 1$), each transaction finishes in 101 steps without any conflicts. Already with 10 times as

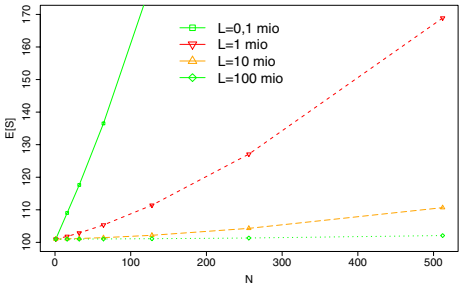


Fig. 4. Mean number of steps $E[S]$ per transaction vs N for different L (and fixed $k = 100$)

many transactional data objects, the performance at $N = 512$ becomes more than acceptable. In this case, transactions with $k = 100$ read/write operations require 110.66 steps on average to finish. With respect to mean number of restarts, we also observed that – in ranges common for applications – increasing L by a factor of 10 decreases $E[R]$ by a factor of around 10 – especially for smaller values of N. For larger N and small L, the reduction in $E[R]$ diminishes.

5 Conclusions

We adapted an approach to model base STM systems to optimistic STM with write buffering and lazy locking. Based on few critical input parameters and amenable to efficient evaluation, the presented analytic model helps us to argue about the performance tradeoffs in optimistic STM – also for design choices realizable only in future implementations. Some rules of thumb shedding light on the complex parameter interactions could be identified. Via discrete-event simulation, especially for small values as encountered for $E[R]$ and with statistical significance, this would be a computationally expensive task. In experiments on existing STM systems (with limited parameter sets) – apart from being time-consuming as well – trends are often also blurred due to implementation specifics and other effects, like varying transaction sizes, etc.

References

1. Herlihy, M., Moss, J.E.B.: Transactional memory: Architectural support for lock-free data structures. In: Proc. 20th Annual Int. Symposium on Computer Architecture (ISCA 1993), pp. 289–300. ACM Press, New York (1993)
2. Larus, J., Rajwar, R.: Transactional Memory. Morgan & Claypool Publishers, San Francisco (2007)
3. Herlihy, M., Luchangco, V., Moir, M., Scherer, W.N.: Software transactional memory for dynamic-sized data structures. In: Proc. 22nd ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC 2003), Boston, MA, USA, pp. 92–101 (2003)
4. Shavit, N., Touitou, D.: Software transactional memory. In: Proc. 14th ACM/SIGACT-SIGOPTS Symposium on Principles of Distributed Computing (PODC 1995), Ottawa, Canada, pp. 204–213 (1995)

5. Saha, B., Adl-Tabatabai, A.R., Hudson, R., Minh, C.C., Hertzberg, B.: McRT-STM: A high-performance software transactional memory system for a multi-core runtime. In: Proc. 11th ACM SIGPLAN Symposium of Principles and Practice of Parallel Programming (PPoPP 2006), New York, NY, USA, pp. 187–197 (2006)
6. Hammond, L., Carlstrom, B.D., Wong, V., Hertzberg, B., Chen, M., Kozyrakis, C., Olukotun, K.: Programming with transactional coherence and consistency (TCC). In: Proc. 11th Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2004), pp. 1–13. ACM Press, New York (2004)
7. Chuang, W., Narayanasamy, S., Venkatesh, G., Sampson, J., Biesbrouck, M.V., Pokam, G., Colavin, O., Calder, B.: Unbounded page-based transactional memory. In: Proc. 12th Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2006), Boston, MA, USA, pp. 347–358 (2006)
8. Ananian, C.S., Asanovic, K., Kuszmaul, B.C., Leiserson, C.E., Lie, S.: Unbounded transactional memory. In: Proc. 11th Int. Symposium on High-Performance Computer Architecture (HPCA 2005), Washington, DC, USA, pp. 316–327. IEEE Computer Society, Los Alamitos (2005)
9. Moore, K.E., Bobba, J., Moravan, M.J., Hill, M.D., Wood, D.A.: LogTM: Log-based transactional memory. In: Proc. 12th Int. Symposium on High-Performance Computer Architecture (HPCA 2006), Washington, DC, USA, pp. 254–265. IEEE Computer Society, Los Alamitos (2006)
10. Damron, P., Fedorova, A., Lev, Y., Luchangco, V., Moir, M., Nussbaum, D.: Hybrid transactional memory. In: Proc. 12th Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2006), Boston, MA, USA, pp. 336–346 (2006)
11. Heindl, A., Pokam, G., Adl-Tabatabai, A.R.: An analytical performance model of software transactional memory. In: Proc. IEEE Int. Symposium on Performance Analysis of Systems and Software (ISPASS 2009), Boston, MA, USA (2009)
12. Heindl, A., Pokam, G.: An analytic framework for performance modeling of software transactional memory. Journal of Computer Networks (accepted for publication 2009)
13. von Praun, C., Bordawekar, R., Cascaval, C.: Modeling optimistic concurrency using quantitative dependence analysis. In: Proc. 13th ACM SIGPLAN Symposium of Principles and Practice of Parallel Programming (PPoPP 2008), Salt Lake City, Utah, USA, pp. 185–196 (2008)
14. Zilles, C., Rajwar, R.: Implications of false conflict rate trends for robust software transactional memory. In: Proc. IEEE Int. Symposium on Workload Characterization (IISWC 2007), Boston, MA, USA (2007)
15. Marathe, V.J., Spear, M.F., Heriot, C., Acharya, A., Eisenstat, D., Scherer, W.N., Scott, M.L.: Lowering the overhead of nonblocking software transactional memory. In: Proc. 1st ACM SIGPLAN Workshop on Languages, Compilers, and Hardware Support for Transactional Computing (TRANSACT 2006), Ottawa, Canada (2006)
16. McKinney, E.: Generalized birthday problem. American Mathematical Monthly 73, 385–387 (1966)
17. Kemeny, J.G., Snell, J.L.: Finite Markov Chains. Springer, Heidelberg (1976)
18. Maplesoft: Maple 12. Tool for mathematics and modeling, Waterloo, Ontario, Canada (2009), http://www.maplesoft.com/
19. Heindl, A., Pokam, G.: Modeling software transactional memory with AnyLogic. In: Proc. 2nd Int. Conf. on Simulation Tools and Techniques (SIMUTools 2009), Rome, Italy (2009)
20. XJ-Technologies: Anylogic 6.2.2. Multi-method simulation software, Petersburg, Russian Federation (2009), http://www.xjtek.com/
21. Minh, C.C., Chung, J.W., Kozyrakis, C., Olukotun, K.: STAMP: Stanford Transactional Applications for Multi-Processing. In: Proc. IEEE Int. Symposium on Workload Characterization (IISWC 2008), Seattle, WA, USA, pp. 35–46 (2008)

Optimal Adaptive Inspection Planning Process in Service of Fatigued Aircraft Structures

Konstantin Nechval[1], Nicholas Nechval[2], Gundars Berzinsh[2], Maris Purgailis[2], Uldis Rozevskis[2] , and Vladimir Strelchonok[2]

[1] Transport and Telecommunication Institute, Applied Mathematics Department, Lomonosov Street 1, LV-1019 Riga, Latvia
konstan@tsi.lv
[2] University of Latvia, Mathematical Statistics Department, Raina Blvd 19, LV-1050 Riga, Latvia
{Nicholas Nechval,Gundars Berzinsh,Maris Purgailis, Uldis Rozevskis,Vladimir Strelchonok}nechval@junik.lv

Abstract. In this paper, a control theory is used for planning inspections in service of fatigue-sensitive aircraft structure components under crack propagation. One of the most important features of control theory is its great generality, enabling one to analyze diverse systems within one unified framework. A key idea, which has emerged from this study, is the necessity of viewing the process of planning in-service inspections as an adaptive control process. Adaptation means the ability of self-modification and self-adjustment in accordance with varying conditions of environment. The adaptive control of inspection planning process in service of fatigued aircraft structures differs from ordinary stochastic control of inspection planning process in that it attempts to reevaluate itself in the light of uncertainties in service of aircraft structures as they unfold and change. Thus, a catastrophic accident during flight can be avoided.

Keywords: Aircraft, fatigue crack, inspection, optimal adaptive planning.

1 Introduction

In spite of decades of investigation, fatigue response of materials is yet to be fully understood. This is partially due to the complexity of loading at which two or more loading axes fluctuate with time. Examples of structures experiencing such complex loadings are automobile, aircraft, off-shores, railways and nuclear plants. Fluctuations of stress and/or strains are difficult to avoid in many practical engineering situations and are very important in design against fatigue failure. There is a worldwide need to rehabilitate civil infrastructure. New materials and methods are being broadly investigated to alleviate current problems and provide better and more reliable future services.

While most industrial failures involve fatigue, the assessment of the fatigue reliability of industrial components being subjected to various dynamic loading situations is one of the most difficult engineering problems. This is because material

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 354–369, 2009.

degradation processes due to fatigue depend upon material characteristics, component geometry, loading history and environmental conditions.

According to many experimental results and field data, even in well-controlled laboratory conditions under constant amplitude loading, crack growth results usually show a considerable statistical variability.

Fatigue is one of the most important problems of aircraft arising from their nature as multiple-component structures, subjected to random dynamic loads. The analysis of fatigue crack growth is one of the most important tasks in the design and life prediction of aircraft fatigue-sensitive structures (for instance, wing, fuselage) and their components (for instance, aileron or balancing flap as part of the wing panel, stringer, etc.).

Airworthiness regulations require proof that aircraft can be operated safely. This implies that critical components must be replaced or repaired before safety is compromised. For guaranteeing safety, the structural life ceiling limits of the fleet aircraft are defined from three distinct approaches: Safe-Life, Fail-Safe, and Damage-Tolerant approaches.

The common objectives to define fleet aircraft lives by the three approaches are to ensure safety while at the same time reducing total ownership costs. Although the objectives of the three approaches are the same, they vary with regard to the fundamental definition of service life.

The Safe-Life approach is based on the concept that significant damage, i.e. fatigue cracking, will not develop during the service life of a component. The life is initially determined from fatigue test data and calculations using a cumulative damage ''law''. Then the design Safe-Life is obtained by applying a safety factor. When the service life equals the design Safe-Life the component must be replaced.

The Fail-Safe approach assumes initial damage as manufactured and its subsequent growth during service to detectable crack sizes or greater. Service life in Fail-Safe structures can thus be defined as the time to a service detectable damage.

However, there are two major drawbacks to the Safe-Life and Fail-Safe approaches: (1) components are taken out of service even though they may have substantial remaining lives; (2) despite all precautions, cracks sometimes occur prematurely. These facts led the Airlines to introduce the Damage Tolerance approach.

The Damage Tolerance approach is based on the concept that damage can occur and develop during the service life of a component. Also, it assumes that cracks or flaws can be present in new structures. Safety is obtained from this approach by the requirements that either (1) any damage will be detected by routine inspection before it results in a dangerous reduction of the static strength (inspectable components), or (2) initial damage shall not grow to a dangerous size during the service life (non-inspectable components). For Damage Tolerance approach to be successful it must be possible to:

- Define either a minimum crack size that will not go undetected during routine inspections, or else an initial crack size, nominally based on pre-service inspection capability.
- Predict crack growth during the time until the next inspection or until the design service life is reached.

An adjunct to Damage Tolerance is Durability analysis. This is an economic life assessment for components that are not safety-critical. The prediction of crack growth is similar to that for Damage Tolerance approach, except that a much smaller initial crack size is used.

2 Stochastic Modelling

To capture the statistical nature of fatigue crack growth, different stochastic models have been proposed in the literature. Some of the models are purely based on direct curve fitting of the random crack growth data, including their mean value and standard deviation (Bogdanoff and Kozin [1]). These models, however, have been criticized by other researchers, because less crack growth mechanisms have been included in them. To overcome this difficulty, many probabilistic models adopted the crack growth equations proposed by fatigue experimentalists, and randomized the equations by including random factors into them (Lin and Yang [2]; Yang et al. [3]; Yang and Manning [4]; Nechval et al. [5-7]; Straub and Faber [8]). The random factor may be a random variable, a random process of time, or a random process of space. It then creates a random differential equation. The solution of the differential equation reveals the probabilistic nature as well as the scatter phenomenon of the fatigue crack growth. To justify the applicability of the probabilistic models mentioned above, fatigue crack growth data are needed. However, it is rather time-consuming to carry out experiments to obtain a set of statistical meaningful fatigue crack growth data. To the writers' knowledge, there are only a few data sets available so far for researchers to verify their probabilistic models. Among them, the most famous data set perhaps is the one produced by Virkler et al. [9] more than twenty years ago. More frequently used data sets include one reported by Ghonem and Dore [10]. Itagaki and his associates have also produced some statistically meaningful fatigue crack growth data, but have not been mentioned very often (Itagaki et al. [11]). In fact, many probabilistic fatigue crack growth models are either lack of experimental verification or just verified by only one of the above data sets. It is suspected that a model may explain a data set well but fail to explain another data set. The universal applicability of many probabilistic models still needs to be checked carefully by other available data sets.

Many probabilistic models of fatigue crack growth are based on the deterministic crack growth equations. The most well known equation is

$$\frac{\mathrm{d}a(t)}{\mathrm{d}t} = q(a(t))^b \tag{1}$$

in which q and b are constants to be evaluated from the crack growth observations. The independent variable t can be interpreted as either stress cycles, flight hours, or flights depending on the applications. It is noted that the power-law form of $q(a(t))^b$ at the right hand side of (1) can be used to fit some fatigue crack growth data appropriately and is also compatible with the concept of Paris–Erdogan law (Paris and Erdogan [12]). The service time for a crack to grow from size $a(t_0)$ to $a(t)$ (where $t > t_0$) can be found by performing the necessary integration

$$\int_{t_0}^{t} dt = \int_{a(t_0)}^{a(t)} \frac{\mathrm{d}v}{qv^b} \tag{2}$$

to obtain

$$t - t_0 = \frac{[a(t_0)]^{-(b-1)} - [a(t)]^{-(b-1)}}{q(b-1)}. \tag{3}$$

For the particular case (when b=1), it can be shown, using Lopital's rule, that

$$t - t_0 = \frac{\ln[a(t)/a(t_0)]}{q}. \tag{4}$$

Thus, we have obtained the Exponential model

$$a(t) = a(t_0)e^{q(t-t_0)}. \tag{5}$$

The Exponential model is quite often used for calculation of growth of population/bacteria etc. The basic equation of it is

$$P_t = P_0 e^{rt}. \tag{6}$$

Rewrite (4) as

$$\tau_{j+1} - \tau_j = \frac{\ln[a(\tau_{j+1})/a(\tau_j)]}{q}, \quad j=0, 1, \ldots . \tag{7}$$

where τ_j is the time of the jth in-service inspection of the aircraft structure component, $a(\tau_j)$ is the fatigue crack size detected in the component at the jth inspection.

It is assumed, in this paper, that the parameter q is a random variable, i.e. $q \equiv Q$, which can take values within a finite set $\{q^{(1)}, q^{(2)}, \ldots, q^{(r)}\}$. However, in order to simplify the computation, at first we consider the case when only two values are chosen. Assume that, at any sampling time instant, the random parameter Q takes on two values, $q^{(1)}$ and $q^{(2)}$, with probabilities p and 1-p, respectively, and that the value of the probability p is not known. It takes on two values p_1 and p_2 with a priori probability ξ and $1-\xi$, respectively. Now (7) can be rewritten as

$$x_{j+1} = x_j + Qu_j, \quad j=0, 1, \ldots , \tag{8}$$

where

$$x_j = \ln[a(\tau_j)], \tag{9}$$

$$u_j = \tau_{j+1} - \tau_j \tag{10}$$

represents the interval between the jth and (j+1)th inspections.

3 Terminal-Control Problem

Let us suppose that a fatigue-sensitive component such as, say, upper longeron [13] (Fig. 1) has been found cracked on one aircraft at the time τ_0. The detectable crack length is $a_0=a(\tau_0)$. The maximum allowable crack length is $a^\bullet$ =4.75 mm (Fig. 2).

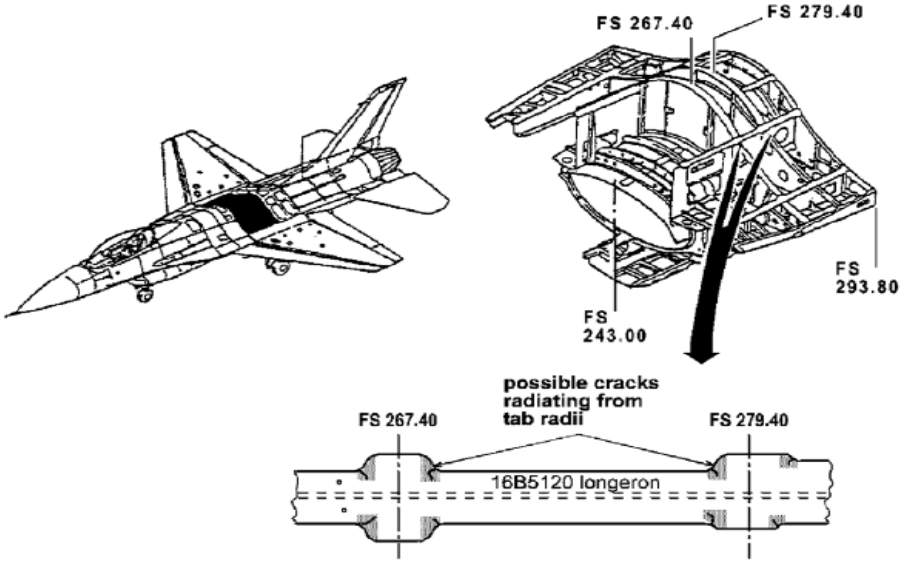


Fig. 1. Inspection points of the upper longeron of RNLAF F-16 aircraft

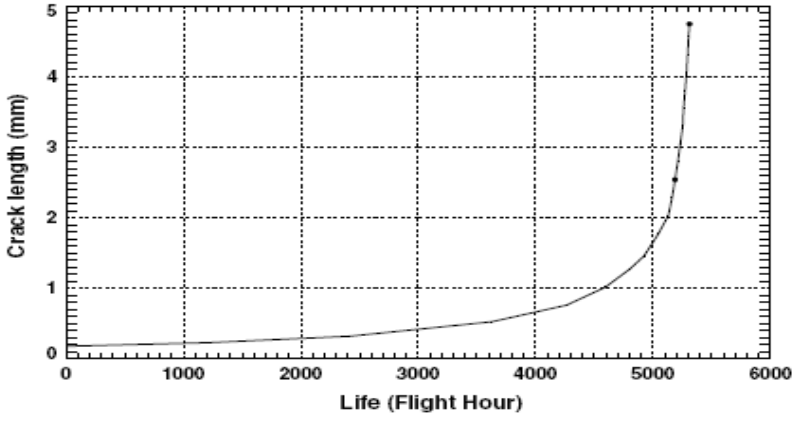


Fig. 2. RNLAF longeron mean crack growth curve. Functional impairment at 5310 flight hours. (assumed initial crack = 0.178 mm; critical crack length = 4.75 mm).

We plan to carry out N in-service inspections of the above component and are in need to assign intervals, u_0, u_1, …, u_{N-1}, between sequential inspections so that the performance index

$$I = E\left\{(x^{\bullet} - x_N)^2\right\}, \tag{11}$$

where $x^{\bullet} = \ln(a^{\bullet})$, is minimized.

4 Optimal Adaptive Inspection Planning Process

The design is initiated with the determination of the a posteriori probabilities

$$\xi_{1:q^{(1)}} = \Pr\{p = p_1 \mid Q = q^{(1)}\} \tag{12}$$

and

$$\xi_{1:q^{(2)}} = \Pr\{p = p_1 \mid Q = q^{(2)}\}. \tag{13}$$

By the Bayes theorem, it is found that

$$\begin{aligned}\xi_{1:q^{(1)}} &= \frac{\Pr\{p = p_1\}\Pr\{Q = q^{(1)} \mid p = p_1\}}{\Pr\{p = p_1\}\Pr\{Q = q^{(1)} \mid p = p_1\} + \Pr\{p = p_2\}\Pr\{Q = q^{(1)} \mid p = p_2\}\}} \\ &= \frac{\xi\, p_1}{\xi\, p_1 + (1-\xi)p_2}\end{aligned} \tag{14}$$

and

$$\begin{aligned}\xi_{1:q^{(2)}} &= \frac{\Pr\{p = p_1 \Pr\{Q = q^{(2)} \mid p = p_1\}}{\Pr\{p = p_1\}\Pr\{Q = q^{(2)} \mid p = p_1\} + \Pr\{p = p_2\}\Pr\{Q = q^{(2)} \mid p = p_2\}\}} \\ &= \frac{\xi(1-p_1)}{\xi(1-p_1) + (1-\xi)(1-p_2)}.\end{aligned} \tag{15}$$

Let the minimum of I be denoted by $f_N(x_0, \xi)$, where

$$x_0 = \ln(a_0), \tag{16}$$

a_0 is the initial crack length detected in the component. The minimum of I is a function of x_0 and the a priori probability ξ, and is given by

$$f_N(x_0, \xi) = \min_{(u_0, u_1, \ldots, u_{N-1})} E\{(x^{\bullet} - x_N)^2\}. \tag{17}$$

At any sampling instant $j + 1$, x_{j+1} takes on two values:

$$x_{j+1}^{(1)} = x_j + q^{(1)} u_j \tag{18}$$

and

$$x_{j+1}^{(2)} = x_j + q^{(2)} u_j. \tag{19}$$

For $j = 0$, x_1 takes on the value

$$x_1^{(1)} = x_0 + q^{(1)}u_0 \quad \text{with probability } p_0 \tag{20}$$

and the value

$$x_1^{(2)} = x_0 + q^{(2)}u_0 \quad \text{with probability } 1-p_0, \tag{21}$$

where p_0 is the expected value of p and is given by

$$p_0 = \xi\, p_1 + (1-\xi)\, p_2\,. \tag{22}$$

Hence, for N=1,

$$f_1(x_0,\xi) = \min_{u_0} E\{(x^\bullet - x_1)^2\} = \min_{u_0}\left(p_0(x^\bullet - x_1^{(1)})^2 + (1-p_0)(x^\bullet - x_1^{(2)})^2\right). \tag{23}$$

For $N \geq 2$, invoking the principle of optimality yields

$$f_N(x_0,\xi) = \min_{u_0}\left(p_0 f_{N-1}(x_1^{(1)}, \xi_{1:q^{(1)}}) + (1-p_0) f_{N-1}(x_1^{(2)}, \xi_{1:q^{(2)}})\right), \tag{24}$$

where $\xi_{1:q^{(1)}}$, $\xi_{1:q^{(2)}}$, $x_1^{(1)}$, and $x_1^{(2)}$ are defined in (14), (15), (20), and (21), respectively. As a result of the first decision, the process will be transformed to one of the two possible states $x_1^{(1)}$ or $x_1^{(2)}$ with probability p_0 or $1-p_0$. If the process moves to state $x^{(1)}$, the a posteriori probability $\xi_{1:q^{(1)}}$ is computed. If the process moves to state $x^{(2)}$, the a posteriori probability $\xi_{1:q^{(2)}}$ is determined.

In a one-stage process, the optimum decision is found by differentiating (23) with respect to u_0 and equating the partial derivative to zero. This leads to

$$p_0 q^{(1)}(x^\bullet - x_0 - q^{(1)}u_0) + (1-p_0)q^{(2)}(x^\bullet - x_0 - q^{(2)}u_0) = 0. \tag{25}$$

Hence

$$u_0 = \frac{E\{Q\}}{E\{Q^2\}}(x^\bullet - x_0), \tag{26}$$

where

$$E\{Q\} = p_0 q^{(1)} + (1-p_0)q^{(2)} \tag{27}$$

and

$$E\{Q^2\} = p_0[q^{(1)}]^2 + (1-p_0)[q^{(2)}]^2 \tag{28}$$

are functions of ξ. By defining

$$E_i\{Q\} = p_i q^{(1)} + (1-p_i)q^{(2)}\,, \quad i=1, 2, \tag{29}$$

it can readily be shown that $E\{Q\}$ can be written as

$$E\{Q\} = \xi\, E_1\{Q\} + (1-\xi)E_2\{Q\}. \tag{30}$$

Similarly, by defining

$$E_i\{Q^2\} = p_i[q^{(1)}]^2 + (1-p_i)[q^{(2)}]^2\,, \quad i=1, 2, \tag{31}$$

$E\{Q^2\}$ can be expressed in terms of ξ as

$$E\{Q^2\} = \xi\, E_1\{Q^2\} + (1-\xi)E_2\{Q^2\}. \tag{32}$$

The minimum for the one-stage process is given by

$$f_1(x_0,\xi) = G_1(\xi)(x^\bullet - x_0)^2, \tag{33}$$

where

$$G_1(\xi) = 1 - \frac{E^2\{Q\}}{E\{Q^2\}}. \tag{34}$$

By defining

$$h_0(\xi) = \frac{E\{Q\}}{E\{Q^2\}} \tag{35}$$

the optimum decision u_0 may be written as

$$u_0 = h_0(\xi)(x^\bullet - x_0). \tag{36}$$

It can be shown by mathematical induction that

$$f_k(x_0,\xi) = G_k(\xi)(x^\bullet - x_0)^2. \tag{37}$$

In view of (37),

$$f_k(x_1^{(1)},\xi_{1:q^{(1)}}) = G_k(\xi_{1:q^{(1)}})(x^\bullet - x_0 - q^{(1)}u_0)^2, \tag{38}$$

$$f_k(x_1^{(2)},\xi_{1:q^{(2)}}) = G_k(\xi_{1:q^{(2)}})(x^\bullet - x_0 - q^{(2)}u_0)^2. \tag{39}$$

The minimum for a (k+1)-stage process is

$$f_{k+1}(x_0,\xi) = \min_{u_0}\left(p_0 G_k(\xi_{1:q^{(1)}})(x^\bullet - x_0 - q^{(1)}u_0)^2 + (1-p_0)G_k(\xi_{1:q^{(2)}})(x^\bullet - x_0 - q^{(2)}u_0)^2\right),$$
$$k = 1, 2, \ldots, N-1. \tag{40}$$

From this recurrence relationship it is found that the optimum decision is given by

$$u_0 = h_k(\xi)(x^\bullet - x_0), \tag{41}$$

where

$$h_k(\xi) = \frac{E\{QG_k(\xi_{1:Q})\}}{E\{Q^2G_k(\xi_{1:Q})\}}, \tag{42}$$

$$E\{QG_k(\xi_{1:Q})\} = \xi\, E_1\{QG_k(\xi_{1:Q})\} + (1-\xi)E_2\{QG_k(\xi_{1:Q})\}, \tag{43}$$

$$E_i\{QG_k(\xi_{1:Q})\} = p_i\, q^{(1)}G_k(\xi_{1:q^{(1)}})\} + (1-p_i)q^{(2)}G_k(\xi_{1:q^{(2)}})\},\ i=1, 2, \tag{44}$$

$$E\{Q^2G_k(\xi_{1:Q})\} = \xi\, E_1\{Q^2G_k(\xi_{1:Q})\} + (1-\xi)E_2\{Q^2G_k(\xi_{1:Q})\}, \tag{45}$$

$$E_i\{Q^2G_k(\xi_{1:Q})\} = p_i\,[q^{(1)}]^2G_k(\xi_{1:q^{(1)}})\} + (1-p_i)[q^{(2)}]^2G_k(\xi_{1:q^{(2)}})\},\ i=1, 2. \tag{46}$$

From (40) and (41) it follows that

$$f_{k+1}(x_0,\xi) = G_{k+1}(\xi)(x^\bullet - x_0)^2, \tag{47}$$

where

$$G_{k+1}(\xi) = E\{G_k(\xi_{1:Q})\} - \frac{E^2\{QG_k(\xi_{1:Q})\}}{E\{Q^2G_k(\xi_{1:Q})\}}, \tag{48}$$

$$E\{G_k(\xi_{1:Q})\} = \xi\, E_1\{G_k(\xi_{1:Q})\} + (1-\xi)E_2\{G_k(\xi_{1:Q})\}, \tag{49}$$

$$E_i\{G_k(\xi_{1:Q})\} = p_iG_k(\xi_{1:q^{(1)}})\} + (1-p_i)G_k(\xi_{1:q^{(2)}})\}, \quad i=1, 2, \tag{50}$$

Equations (33), (34), (47), and (48) are recurrence relationships with which it is possible to evaluate the minimum for an N-stage process $f_N(x_0,\xi)$.

With the initial state x_0 and initial information ξ, the first optimum decision is

$$u_0 = h_{N-1}(\xi)(x^\bullet - x_0), \tag{51}$$

where $h_{N-1}(\xi)$ is evaluated from (42) to (46) and (48) to (50), with $k = N-1$. The second optimum decision should be made after observation of the random variable Q in the first decision stage. If it is observed that $Q = q^{(1)}$, the a posteriori probability $\xi_{1:q^{(1)}}$ and the new state

$$x_1^{(1)} = x_0 + q^{(1)}u_0 \tag{52}$$

are used as the initial information for the remaining $N-1$ stages. The second optimum decision can be determined in similar manner and is given by

$$u_1 = h_{N-2}(\xi_{1:q^{(1)}})(x^\bullet - x_1^{(1)}). \tag{53}$$

If the observed value of Q after the first decision is $q^{(2)}$, the a posteriori probability $\xi_{1:q^{(2)}}$ and the new state

$$x_1^{(2)} = x_0 + q^{(2)}u_0 \tag{54}$$

are used as the initial information and the initial state for the remaining $N-1$ stages. The second optimum decision is then given by

$$u_1 = h_{N-2}(\xi_{1:q^{(2)}})(x^\bullet - x_1^{(2)}). \tag{55}$$

Thus, after the first inspection, the computer must calculate the a posteriori probability $\xi_{1:q^{(1)}}$ or $\xi_{2:q^{(2)}}$, the new state x_1 and the second optimum decision u_1.

If the observed value of Q after the second decision is $q^{(1)}$, the a posteriori probability $\xi_{2:q^{(1)}}$ and the new state

$$x_2^{(1)} = x_1 + q^{(1)}u_1 \tag{56}$$

are used as the initial information and the initial state for the remaining N - 2 stages, in particular, to determine the third optimum decision

$$u_2 = h_{N-3}(\xi_{2:q^{(1)}})(x^\bullet - x_2^{(1)}) \,. \tag{57}$$

In (56), if $x_1 = x_1^{(1)}$, then a posteriori probability is $\xi_{1:q^{(1)}}$ and u_1 is given by (53); and if $x_1 = x_1^{(2)}$, then a posteriori probability is $\xi_{1:q^{(2)}}$ and u_1 is given by (55).

If the observed value of Q after the second decision is $q^{(2)}$, the a posteriori probability $\xi_{2:q^{(2)}}$ and the new state

$$x_2^{(2)} = x_1 + q^{(2)} u_1 \tag{58}$$

are used to determine the third optimum decision, which is

$$u_2 = h_{N-3}(\xi_{2:q^{(2)}})(x^\bullet - x_2^{(2)}) \,. \tag{59}$$

By repeated observation and computation in the above manner, the optimum-inspection policy (u_0, …, u_{N-1}) for the fatigue-sensitive component, which has been found cracked on one aircraft at the time τ_0, can be determined.

Each new optimum decision is made by using new information resulting from the observation of the random variable Q.

It will be noted that if the probability p is assumed to be known, then the minimum of (11) can be found as follows. Let the minimum of (11) be

$$f_N(x_0) = \min_{(u_0, u_1, \ldots, u_{N-1})} E\{(x^\bullet - x_N)^2\}. \tag{60}$$

For N=1,

$$f_1(x_0) = \min_{u_0} E\{(x^\bullet - x_1)^2\} = \min_{u_0} \left(p(x^\bullet - x_1^{(1)})^2 + (1-p)(x^\bullet - x_1^{(2)})^2 \right)$$

$$= \left(1 - \frac{E^2\{Q\}}{E\{Q^2\}}\right)(x^\bullet - x_0)^2 \tag{61}$$

with

$$u_0 = \frac{E\{Q\}}{E\{Q^2\}}(x^\bullet - x_0), \tag{62}$$

where

$$E\{Q\} = pq^{(1)} + (1-p)q^{(2)} \tag{63}$$

and

$$E\{Q^2\} = p[q^{(1)}]^2 + (1-p)[q^{(2)}]^2 \tag{64}$$

are functions of p.

For $N \geq 2$, the minimum of (11) is

$$f_N(x_0) = \min_{u_0} \left(p f_{N-1}(x_1^{(1)}) + (1-p) f_{N-1}(x_1^{(2)}) \right)$$

$$= \left(1 - \frac{E^2\{Q\}}{E\{Q^2\}}\right)^N (x^\bullet - x_0)^2 \tag{65}$$

with u_0 given by (62).

For an illustration, one of the versions (for N=5) of adaptive minimizing the expected value of the performance index (11) for the upper longeron of RNLAF F-16 aircraft is plotted in Fig. 3.

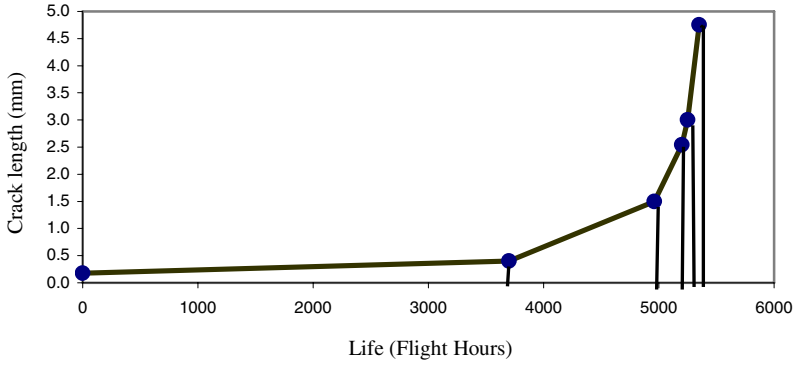


Fig. 3. Inspection schedule version for the upper longeron of RNLAF F-16 aircraft

Fig.4 shows the deterministic inspection requirements [14] for the RNLAF longerons.

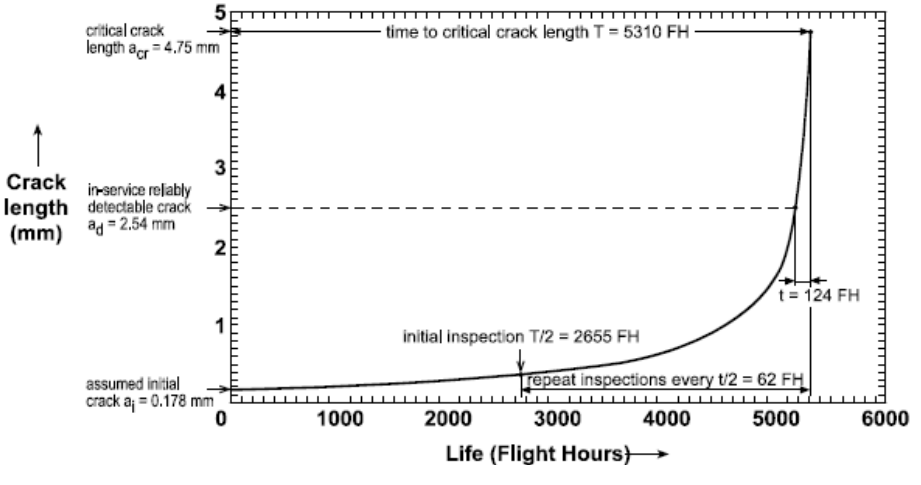


Fig. 4. Deterministic Damage Tolerance inspection requirements for the RNLAF longerons

Now consider the case when the parameter q is a random variable, i.e. $q \equiv Q$, which can take values $q^{(1)}, q^{(2)}, \ldots, q^{(r)}$ with probabilities $p_1, p_2, \ldots, p_r$, respectively, where

$$p_i \geq 0, \quad \sum_{i=1}^{r} p_i = 1. \tag{66}$$

These probabilities are unknown. Suppose the parameter Q is observed n times. Let n_i = number of occurrences of $Q = q^{(i)}$. Clearly

$$\sum_{i=1}^{r} n_i = n. \tag{67}$$

Then, the likelihood function is given by

$$L(n_1, ..., n_r \mid n, q^{(1)}, ..., q^{(r)}, p_1, ..., p_r) = \binom{n}{n_1, ..., n_r} p_1^{n_1} \cdots p_r^{n_r}, \tag{68}$$

a multinomial distribution.

The convenient prior distribution to use over the p_i's is a member of the multidimensional Beta family, i.e.,

$$\xi(p_1, ..., p_r) = \frac{1}{\mathrm{B}(m_1, ..., m_r)} p_1^{m_1 - 1} \cdots p_r^{m_r - 1}, \tag{69}$$

where $\mathrm{B}(m_1, \ldots, m_r)$ is the generalized Beta function defined by

$$\mathrm{B}(m_1, ..., m_r) = \frac{\prod_{i=1}^{r} \Gamma(m_i)}{\Gamma\left(\sum_{i=1}^{r} m_i\right)} = \mathrm{B}\left(m_1, \sum_{i=2}^{r} m_i\right) \mathrm{B}\left(m_2, \sum_{i=3}^{r} m_i\right) \cdots \mathrm{B}(m_{r-1}, m_r), \tag{70}$$

$$\Gamma(m) = \int_0^{\infty} x^{m-1} e^{-x} dx. \tag{71}$$

For a positive integer m, $\Gamma(m) = (m-1)!$

To verify that (69) is in fact a frequency function, we note that

$$p_r = 1 - \sum_{i=1}^{r-1} p_i. \tag{72}$$

Restricting to the case of three random variables ($r-1=3$) for convenience, and recalling that

$$\sum_{i=1}^{3} p_i < 1, \tag{73}$$

we have to show that

$$\frac{1}{\mathrm{B}(m_1, m_2 + m_3 + m_4)\,\mathrm{B}(m_2, m_3 + m_4)\,\mathrm{B}(m_3, m_4)}$$

$$\times \int_0^1 dp_1 \int_0^{1-p_1} dp_2 \int_0^{1-p_1-p_2} p_1^{m_1-1} p_2^{m_2-1} p_3^{m_3-1} (1-p_1-p_2-p_3)^{m_4-1} dp_3 = 1. \tag{74}$$

Using repeatedly the relation

$$\int_0^a p^{m-1}(a-p)^{n-1} dp = a^{m+n-1} \mathrm{B}(n,m), \tag{75}$$

the integral in (74) is readily seen to equal unity. This result is easily generalized to any number of variables. Thus (69) is a frequency function.

From our choice of likelihood function and prior distribution it follows directly that the ensuing posterior distribution will be a new member of the same multidimensional Beta family (a consequence of the judicious choice of the prior family).

The new parameters (m_i'') are easily obtained from the old ones (m_i') and the observed data (n_i) by means of the following rule: $m_i'' = m_i' + n_i$. The prior parameters, ($m_0, \ldots, m_r$) have to be selected. If the decision-maker has prior beliefs it is logical to select the parameters to reflect these.

If we integrate (69) over all p_i except p_j we obtain the marginal prior distribution of p_j,

$$\xi(p_j) = \frac{1}{\mathrm{B}(m_j, \sum_{i \neq j} m_i)} p_j^{m_j - 1} (1-p_j)^{\sum_{i \neq j} m_i - 1}, \quad 0 \le p_j \le 1. \tag{76}$$

The prior probability of $Q = q^{(j)}$ is given by

$$\Pr\{Q = q^{(j)}\} = p_{0j} = \int_0^1 p_j \xi(p_j) dp_j = \frac{m_j}{\sum_{i=1}^r m_i}. \tag{77}$$

Thus, with the determination of the a posteriori distributions

$$\xi_{1q^{(j)}}(p_1, \ldots, p_r \mid Q = q^{(j)}) = \frac{1}{\mathrm{B}(m_1, \ldots, m_j + 1, \ldots, m_r)} p_1^{m_1 - 1} \cdots p_j^{m_j} \cdots p_r^{m_r - 1},$$

$$j = 1, \ldots, r, \tag{78}$$

the marginal a posteriori distributions

$$\xi_{1q^{(j)}}(p_i \mid Q = q^{(j)}) = \frac{1}{\mathrm{B}(m_i, \sum_{s \neq i,j} m_s + m_j + 1)} p_i^{m_i - 1} (1-p_i)^{\sum_{s \neq i,j} m_s + m_j + 1}, \quad 0 \le p_i \le 1, \tag{79}$$

and the a posteriori probabilities

$$p_{1i}(j) = \int_0^1 p_i \xi_{1q^{(j)}}(p_i \mid Q = q^{(j)}) dp_i = \frac{m_i}{\sum_{s=1}^r m_s + 1}, \quad i \neq j; \quad p_{1j}(j) = \frac{m_j + 1}{\sum_{s=1}^r m_s + 1}, \tag{80}$$

we obtain the following.

For N=1, the minimum of (11) is

$$f_1(x_0,\xi)=\min_{u_0} E\{(x^\bullet-x_1)^2\}=G_1(\xi)(x^\bullet-x_1)^2=\left(1-\frac{E^2\{Q\}}{E\{Q^2\}}\right)(x^\bullet-x_0)^2 \tag{81}$$

with

$$u_0=h_0(\xi)(x^\bullet-x_0)=\frac{E\{Q\}}{E\{Q^2\}}(x^\bullet-x_0), \tag{82}$$

where

$$E\{Q\}=\sum_{i=1}^{r} p_{0i}q^{(i)} \tag{83}$$

and

$$E\{Q^2\}=\sum_{i=1}^{r} p_{0i}[q^{(i)}]^2. \tag{84}$$

For $N\geq 2$, the minimum of (11) is

$$f_N(x_0,\xi)=\min_{u_0}\left(\sum_{i=1}^{r} p_{0i}f_{N-1}(x_1^{(i)})\right)$$

$$=G_N(\xi)(x^\bullet-x_0)^2=\left(E\{G_{N-1}(\xi_{1Q})\}-\frac{E^2\{QG_{N-1}(\xi_{1Q})\}}{E\{Q^2G_{N-1}(\xi_{1Q})\}}\right)(x^\bullet-x_0)^2 \tag{85}$$

with

$$u_0=h_{N-1}(\xi)(x^\bullet-x_0)=\frac{E\{QG_{N-1}(\xi_{1Q})\}}{E\{Q^2G_{N-1}(\xi_{1Q})\}}(x^\bullet-x_0), \tag{86}$$

where

$$E\{QG_{N-1}(\xi_{1Q})\}=\sum_{i=0}^{r} p_{0i}q^{(i)}G(\xi_{1q^{(i)}}),\quad E\{Q^2G_{N-1}(\xi_{1Q})\}=\sum_{i=0}^{r} p_{0i}[q^{(i)}]^2G(\xi_{1q^{(i)}}). \tag{87}$$

5 Optimal Number of In-Service Inspections

By plotting $f_N(x_0,\xi)$ versus N the optimal number of in-service inspections N^* can be determined as

$$N^*=\arg\inf_N[c_f f_N(x_0,\xi)+c_N N], \tag{88}$$

where c_f and c_N represent the specified weight coefficients. Fig. 5 illustrates the graphical method of finding the optimal number of in-service inspections.

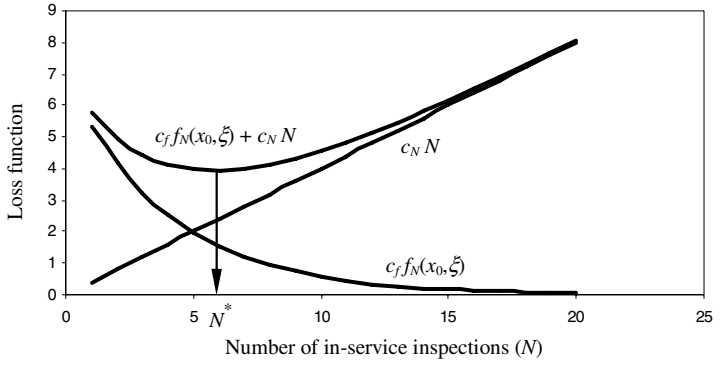


Fig. 5. Graphical method of finding the optimal number N^* of in-service inspections

6 Conclusion

An analytical solution to the terminal-control problem is generally not easy to derive, and numerical procedures should be followed. In this paper, the design of adaptive and learning control processes is considered. The design of such processes is carried out on the basis of the Bayes theorem and the functional-equation approach of dynamic programming. An inspection planning process of cracked aircraft structure component is used to illustrate the design procedure.

Acknowledgments. This research was supported in part by Grant No. 06.1936 and Grant No. 07.2036 from the Latvian Council of Science and the National Institute of Mathematics and Informatics of Latvia.

References

1. Bogdanoff, J.L., Kozin, F.: Probabilistic Models of Cumulative Damage. Wiley, New York (1985)
2. Lin, Y.K., Yang, J.N.: On Statistical Moments of Fatigue Crack Propagation. Engineering Fracture Mechanics 18, 243–256 (1985)
3. Yang, J.N., His, W.H., Manning, S.D.: Stochastic Crack Propagation with Applications to Durability and Damage Tolerance Analyses. Technical Report, Flight Dynamics Laboratory, Wright-Patterson Air Force Base, AFWAL-TR-85-3062 (1985)
4. Yang, J.N., Manning, S.D.: Stochastic Crack Growth Analysis Methodologies for Metallic Structures. Engineering Fracture Mechanics 37, 1105–1124 (1990)
5. Nechval, N.A., Nechval, K.N., Vasermanis, E.K.: Statistical Models for Prediction of the Fatigue Crack Growth in Aircraft Service. In: Varvani-Farahani, A., Brebbia, C.A. (eds.) Fatigue Damage of Materials 2003, pp. 435–445. WIT Press, Southampton (2003)
6. Nechval, N.A., Nechval, K.N., Vasermanis, E.K.: Estimation of Warranty Period for Structural Components of Aircraft. Aviation VIII, 3–9 (2004)

7. Nechval, N.A., Nechval, K.N., Berzinsh, G., Purgailis, M., Rozevskis, U.: Stochastic Fatigue Models for Efficient Planning Inspections in Service of Aircraft Structures. In: Al-Begain, K., Heindl, A., Telek, M. (eds.) ASMTA 2008. LNCS, vol. 5055, pp. 114–127. Springer, Heidelberg (2008)
8. Straub, D., Faber, M.H.: Risk Based Inspection Planning for Structural Systems. Structural Safety 27, 335–355 (2005)
9. Virkler, D.A., Hillberry, B.M., Goel, P.K.: The Statistic Nature of Fatigue Crack Propagation. ASME Journal of Engineering Materials and Technology 101, 148–153 (1979)
10. Ghonem, H., Dore, S.: Experimental Study of the Constant Probability Crack Growth Curves under Constant Amplitude Loading. Engineering Fracture Mechanics 27, 1–25 (1987)
11. Itagaki, H., Ishizuka, T., Huang, P.Y.: Experimental Estimation of the Probability Distribution of Fatigue Crack Growth Lives. Probabilistic Engineering Mechanics 8, 25–34 (1993)
12. Paris, R., Erdogan, F.: A Critical Analysis of Crack Propagation Laws. Journal of Basic Engineering 85, 528–534 (1963)
13. Grooteman, F.: A Stochastic Approach to Determine Lifetimes and Inspection Schemes for Aircraft Components. International Journal of Fatigue 30, 138–149 (2008)
14. Military Specification: Airplane Damage Tolerance Requirements. MIL-A-83444, USAF (1974)

Stochastic Modelling of Poll Based Multimedia Productions

Pietro Piazzolla[1], Marco Gribaudo[1], and Alberto Messina[2]

[1] Dip. di Informatica, Università di Torino
piazzolla.pietro@di.unito.it, marcog@di.unito.it
[2] RAI - Centre for Research and Technological Innovation
a.messina@rai.it

Abstract. The nowadays explosion of new media distribution channels and the new digital tools based production work-flows require an immediate revision of the traditional ways in which media industry makes its business. Leading experts agree in recognizing automation of processes as one of the key for success in this scenario, especially for the potential production costs reduction introduced by it. However, there is a substantial lack of precision in evaluating the overall economical weight of this new business, which is particularly due to the extra complexity introduced by the advent of non-linear consumption paradigms (like the Internet), in which user's feedback have a central importance. This paper represents an attempt to apply performance evaluation techniques to multi-channel productions, and it illustrates how these methods can help in optimizing the evaluation of costs of this kind of processes.

1 Introduction

The nowadays explosion of new media distribution channels and the new production work-flows based on digital computer-based tools requiere an immediate revision of the traditional ways of making business in media industry. Recent market surveys [4] [11] are demonstrating that in a very near future the Internet based multimedia fruition model will undermine the existing one-to-many broadcasting model, thus putting under serious discussion an important sector of the European tertiary economy.

To cope with these upcoming changes, broadcasters have been revolutionizing their point of view, trying to embrace new models into their facilities rather than being routed by them [1] [2]. However the overall economic convenience of these initiatives is still to be fully proved, and the risk is that of making bets rather than plans. Leading experts all agree in recognizing automation of processes as one of the key for success in this scenario, because of the potential production costs reduction introduced by it. More recently, the adoption of tools for intelligent analysis and synthesis of multimedia data are seen as substantial enabling factors in making interactive, multi-channel and multi-purpose productions value-returning [6] [7] [9].

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 370–385, 2009.

However, there is a substantial lack of precision in evaluating the overall economical weight of this new business, which is particularly due to the extra complexity introduced by the advent of non-linear consumption paradigms (like the Internet), in which users' feedback have a central importance. Though modelling users' behaviors, attitudes and profiles would be a key enabling factor in this scenario, very often this is not practically feasible due to a number of technological and ethical limitations. An alternative approach can be found in introducing stochastic modelling. To address these complex problems with a well-grounded and robust approach, a crucial aspect is represented by the ability of performing a correct process modelling, which should include all the relevant stochastic elements. This paper represents an attempt to apply performance evaluation techniques to multi-channel productions, and it illustrates how these methods can help in optimizing the evaluation of costs of this kind of processes.

2 The Considered System

The main idea of this work is to deal with multimedia contents published on a given platform, at regularly recurring intervals. The collection of these contents constitutes a *series*. We can better explain the *series* concept, using the definition of [12], as any "publication in any medium issued under the same title in a succession of discrete parts, usually numbered (or dated) and appearing at regular or irregular intervals with no predetermined conclusion". In our models we will focus only on regular intervals of equal size D.

We assume throughout this paper that an *episode* is the main medial content to be produced by a *deadline*, equal to the interval size D. A medial content is, for example, a single episode of a television series that must be broadcasted by a certain date and time. Another example of main content is an issue of a newscast. Together with the main content, we suppose the making of additional *extra contents*. These extras are, for example, small video packaged from the main content to be broadcasted on the program's web site, or even full spin-off programs to be scheduled during the intervals between the main episodes. More in general we can say that the *extras* are autonomous and independent contents that may be published on different broadcasting platforms. The extras can also be produced by an automatic generation process, however we imagine that their production will require the same resources used for the main production: that is, extras cannot be produced in parallel with the main contents. We imagine the existence of a key point P, that we will call the *poll deadline*, that identifies a time interval between the production of two consecutive episodes, that can be used to gather information to improve the content of the next episode, based on reactions to the previous one. For example, it can correspond to the results obtained by a generic poll system, used to probe the opinion of the audience about it. We imagine the size of this feedback window fixed in length, and equal for all the episodes composing the series. While in the real-world productions, as far as we know, there aren't yet any poll-based tv series to use as an examples, we can think here to the various reality-shows. In these programs, the content (i.e.

the participants) of the next episode will vary considering the audience vote. Other examples came from some theater dramas, during which the audience is requested to choose one of the possible endings, e.g. *Shear Madness*, a very famous American comedy [3]. By some extend, we can also consider as examples the movies alternative endings published within their DVD versions, e.g. *I Am Legend* which DVD includes an alternate ending more similar to that originally intended by the story author, Richard Matheson. In both cases the audience is required to choose between two or more alternatives already produced. This means that between those alternatives there is not necessarily the public's most favored one. Moreover, all the alternatives published were produced and thus were a cost for the overall production of that movie.

We also assume that, if the production is guided by the poll results, is possible to produce episodes more matching with the audience expectancies, and we are able to estimate this gain. Exploiting the effects of the poll system can become a powerful retention tool for expanding and consolidating the audience during the series production [8] [10]. With the help of such a tool is possible to create a product matching with the audience expectancies without having to produce unnecessary contents. If we consider a newscast, the poll can be interpreted as the possibility to wait for the latest events to report. We assume that the extra contents are made in a time frame that starts after the completion of the current main content and finishes at the end of the next episode poll. That is, their production must finish strictly before the starting of the production of the next episode poll dependent content. The extras exceeding this deadline in their making will not be considered.

As mentioned in the Section 1, there is a lack of precision in evaluating the economical impact of a production influenced by a non-linear consumption paradigm. In our models we will use a common metric that we will generically address as *quality*: an abstract measurement of such a profit. This matric may be used to express the expected overall income of the production, either in terms of public share or financial returns. By using the word *abstract* we mean that while this profit cannot be exactly calculated, we can nonetheless describe its general behavior. We will consider this metric to be *additive*: that is that the total quality can be obtained by summing up the qualities coming from the different parts composing an episode. The considered system will be analyzed using three models of increasing complexity. All the models will share a common measurement of the quality of the poll, described in Section 3.

3 The Poll Model

As introduced in Section 2, we imagine a poll deadline P, that defines the time interval that might be used to gather information that can be used to improve the quality of the next episode. This information can came either from users' feedback (as in the case of reality shows), or from the chance of considering "breaking-news" (as in the case of newscasts).

In case of users' feedback, we call R a continuous positive random variable that describes the time required by a user to answer the poll and provide her/his

feedback. Suppose that we expect N_u users to give their feedback. We call $S(P)$ a discrete random variable that counts the number of answers received before the deadline P. Users can be considered independent, and thus it can be defined as:

$$S(P) = \text{Binom}(N_u, R(P)) \tag{1}$$

where $\text{Binom}(N, p)$ denotes a Binomial distribution with population parameter N and probability parameter p, and $R(t)$ denotes the c.d.f. of random variable R.

Let us call $q_F(n)$ a multiplicative factor that can increase or decrease the total quality of a production, when feedback from n users out of the N_u population is received. $q_F(n)$ should be minimum when $n = 0$ (i.e.: no feedback has been received, and thus taken into account), and maximum when $n = N$ (i.e.: when the feedback came from all the users that has been considered). We can call $\bar{Q}_F(P)$ the average expected feedback quality, when the deadline is set to P. It can be computed as:

$$\bar{Q}_F(P) = \sum_{i=0}^{N_u} \Pr\{S(P) = i\} \cdot q_F(i) \tag{2}$$

For the breaking news example, we imagine to have $\bar{Q}_F(P)$ computed from statistics regarding the probability of having an important event that must be considered before P.

4 The Basic Model

In this section we discuss the basic version of the proposed model. Here we assume the making of a single main content whose production starts right after the poll ending. If the main content is finished before the deadline, the remaining time can be used for producing the extra contents. The production of these extra contents can continue even after the deadline, but must finish before the next episode's poll end. This ensures that at least the next episode's poll time is used for extras production. Figure 1 shows a diagram of a possible configuration for this production.

Let us recall that D defines the episode deadline, and P the poll window. Let us call M a continuous random variable that defines the time required to finish the main production. If the contents would require more time than available (that is more than $D - P$), the production will be forced to finish within the deadline, reducing its quality.

We call $q_M(t)$ the quality of the production that can be achieved by the shortening of a t time. If the production can finish before the deadline, then $t \leq 0$. Otherwise, if the production would have required $D - P + \alpha$ time, then $t = \alpha > 0$, since it must have been shortened of α in order to fit the $D - P$ slot. Therefore we expect $q_M(t)$ being at its maximum value for $t < 0$, and we expect $q_M(t) \to 0$ for $t \to \infty$. For example, a possible definition of the production quality function can be:

$$q_M(t) = \begin{cases} q_{\max} & \text{for } t \leq 0 \\ q_{\max}\, e^{-\left(\frac{t}{q_s}\right)} & \text{for } t > 0 \end{cases} \tag{3}$$

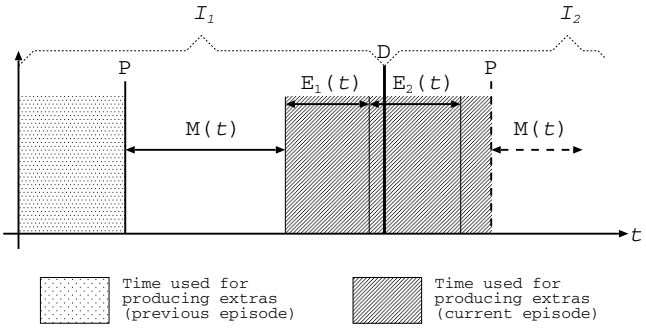


Fig. 1. The basic model

where $q_{\max}$ is the maximum quality that can be achieved, and q_s expresses an exponential decay rate at which the quality decreases as the production is shortened. We can then compute the average quality of the main production content $\bar{Q}_M(P)$ as:

$$\bar{Q}_M(P) = \int_0^\infty m(t) q_M(t - D + P)\ dt \tag{4}$$

where $m(t)$ is the p.d.f. of distribution M. In order to simplify the presentation in Equation (4), and in all subsequent equations, we have not shown the dependence on D.

We imagine that the time wasted waiting for the polling, and the time left before the deadline can be used to produce extra contents. We call E_i a random variable that describes the time required to produce the i-th extra content. We imagine that we put a limit N_e to the number of extra contents that can be produced, and that all the E_i are independent. We define with $N_E(t)$ a random variable that describes the the number of extra contents that can be produced if the time left available to the production of extra contents is t. We have that:

$$\Pr\{N_E(t) = n\} = \begin{cases} \Pr\{E_1 > t\} & n = 0 \\ \displaystyle\int_0^t e_n(s) \Pr\{E_{n+1} > t - s\}\ ds & 0 < n < N_e \\ \Pr\{\displaystyle\sum_{i=1}^{N_e} E_i \leq t\} & n = N_e \end{cases} \tag{5}$$

where $e_n(s)$ is the p.d.f. of the random variable $S_n = \sum_{i=1}^n E_i$. We can use $N_E(t)$ to compute $N_{ED}|P$, the distribution of the number of extra contents, given a deadline P. In particular, we have that:

$$\begin{aligned} \Pr\{N_{ED} = n|P\} = \Pr\{N_E(P) = n\} \Pr\{M \geq D - P\} + \\ + \int_0^{D-P} \Pr\{N_E(D - s) = n\}\ m(s)\ ds \end{aligned}$$

The second term of the r.h.s. of the previous equation takes into account the fact that if the main production can finish earlier than the deadline (at time s) then all the remaining time ($D - s$) can be used to produce extra contents. The first term instead takes into account the cases where the production time is truncated at $D - P$ to respect the deadline, leaving only P (the time before the arrival of the results of the next poll) for extra contents creation.

If we call $q_e(i)$ the quality that we can obtain if i extra contents are produced, then we can compute the average quality of the extra contents $\bar{Q}_E(P)$ as:

$$\bar{Q}_E(P) = \sum_{i=0}^{N_e} \Pr\{N_{ED}(t) = i|P\} q_e(i) \tag{6}$$

For example a simple possible definition for $q_e(i)$ can be:

$$q_e(i) = i \cdot \alpha_e \tag{7}$$

where α_e is a constant that describes the quality gained for each extra content. We can use the previous definitions to compute the average production quality $\bar{Q}_1(P)$, given a deadline P, as:

$$\bar{Q}_1(P) = \bar{Q}_M(P)\bar{Q}_F(P) + \bar{Q}_E(P) \tag{8}$$

We can use Equation (8) to find the optimal P to achieve the highest possible quality.

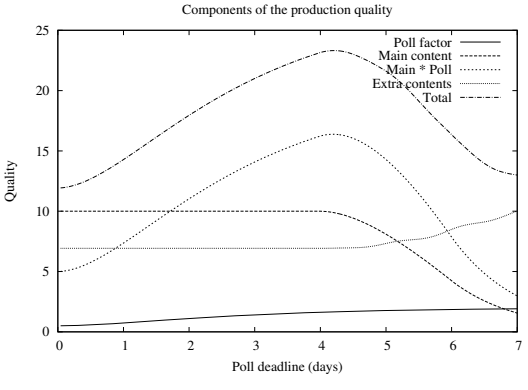


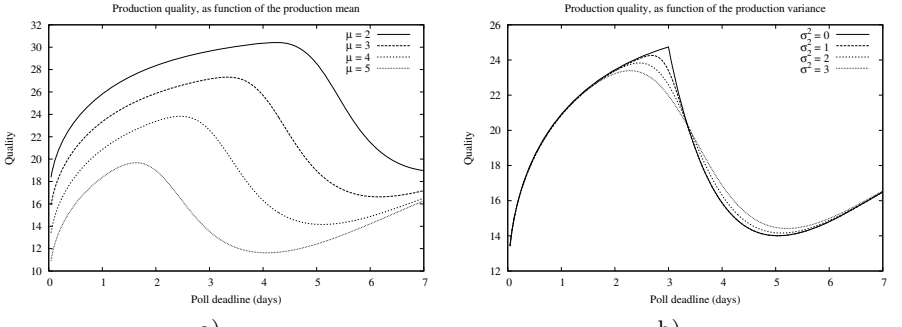
Fig. 2. The quality components

An example of the results we can obtain by this model is shown in Figure 2, where the elements composing the quality are visible. In that experiment, we used a seven days deadline. We expected the poll answers to arrive following an exponential distribution with mean of 2 days. The time required to complete the main content follows an uniform distribution between 1 and 3 days. We also

assumed a maximum of eight extra content to be produced, and their distribution is a gaussian one with a mean of 1 day and variance of 0.5 days.

The main production quality followed Equation (3) with $q_{\max} = 10$ and $q_s = 1$ day(s). Also, the poll quality function was set to $q_F(n) = \frac{1}{2} + \frac{3}{2}\sqrt{\frac{n}{N_u}}$ with $N_u = 100$. $q_e(i)$ follows Equation (7) with $\alpha_e = 2.5$.

It seems quite clear how the poll factor is influencing the main production quality: the more poll information obtained: if all the expected people responds to the poll, the main content quality is doubled. If none of the expected people answers the poll, then the main content quality is halved. This is because we assume that the main content will be made coherently with the audience expectancies. In the same figure is also important to notice that the quality of the main production decrease approaching the deadline. This is because the less probability we have to complete the main content, the less content quality we can obtain. About the extra, the more we are distant from the end of the poll, the more extra we are able to produce because of the time at our disposal. The quality of the extra is proportional to the number of the extras produced, which in turn are proportional to the time available before the start of the next episode production. This time quantity became larger as the poll deadline increases, this is due to the fact that with a larger poll deadline there is an higher chance to truncate the making of the main production in favor of extra content generation.



a) b)

Fig. 3. Quality as function of the production a) mean, b) variance

In Figure 3 we consider the main content making being normal distributed, and we studied the whole production quality as a function of both production mean and variance. Also for this example we used a seven day deadline. In the first case *a*) we can observe how the bigger mean value we have, the bigger quality production we gain. It seems to be a good solution to end the poll before the exact mean of the main production because, otherwise, the drop in quality for not completing the main episode is hardly compensated by other factors. For the variance example *b*) we can see how rising the variance influences the maximum obtainable production quality.

5 Adding an Institutional Content

Until here we assumed the main content to be a single entity. However this assumption is quite non-realistic: in most of the productions there are tasks that must be performed, and that are independent from the feedback. For those tasks, is not meaningful to wait until the end of the poll. For this reason, we extend the model by considering the episode composed by two distinct parts. The first one, that we call *institutional*, is the part of the main content not influenced by the poll results. The second one, the *request based* part, is, of course, oriented by the audience. Indeed an episode relying only on the poll results for deciding it's contents can be very hard to produce.

We imagine that the institutional part of the main content might not start immediately after the end of the previous episode. We allow the introduction of a delay W, that might be used to model organizational issues, before the making of the new episode. Please note that during this waiting time it is still possible to continue the previous episode's extra contents.

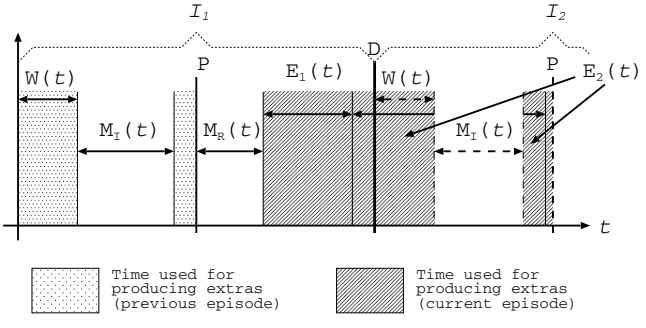


Fig. 4. Institutional and request based contents

Figure 4 shows the schema of this model. In this case, the production time for the extra contents is given by the sum of two different time quantities: the time remaining after finishing the institutional part before the poll end and the time remaining after finishing the request based content before the next episode's poll end. Please note that the production of an extra content can be split in two parts to better use the time available for extras. We give priority to the production of the institutional part. If the institutional content is finished after the poll's end, then request based content will have to wait the end of the former before starting its production. Let's also assume that the W time cannot be longer than D, i.e.: $\Pr\{W > D\} = 0$. The main content distribution M can thus be considered as the sum of two independent distributions: the *institution content distribution* M_I, and the *feedback dependent content distribution* M_R. That is: $M = M_I + M_R$. We also imagine to have two production quality functions $q_{M_I}(t)$ and $q_{M_R}(t)$.

Due to the different assumptions about the time at which each production starts, the average qualities must be computed in two different ways. First note that the average quality of the institutional content $\bar{Q}_{M_I}$ does not depend on the poll time P (since it starts after W, regardless of when the poll ends). If we call $M_W = W + M_I$, and we define $m_W(t)$ as its probability density function, then $\bar{Q}_{M_I}$ can be computed simply computed as:

$$\bar{Q}_{M_I} = \int_{t=0}^{\infty} m_W(t) q_{M_I}(t - D)\ dt \tag{9}$$

Note that since we have supposed that $\Pr\{W > D\} = 0$, we can be certain that at least the production of the institutional content will start before the deadline.

The feedback based content, will be produced starting immediately after P (if the production of the institutional content would have finished earlier than the end of the poll), or after the end of the institutional content whichever it cames last. In particular, we can compute $\bar{Q}_{M_R}(P)$ as:

$$\begin{aligned}\bar{Q}_{M_R}(P) = \Pr\{M_W < P\} \int_0^{\infty} m_R(t) q_{M_R}(t - D + P)\ dt + \\ + \int_P^D \int_0^{\infty} m_W(v) m_R(t) q_{M_R}(t + v - D)\ dt\ dv + \\ + \Pr\{M_W > D\} \int_0^{\infty} m_R(t) q_{M_R}(t)\ dt\end{aligned}$$

where $m_R(t)$ is the p.d.f. of M_I. The first term on the r.h.s. of the previous equation represents the case when the institutional contents finish before the poll, the second one the case when it finishes between P and D, and the third one the case when it finishes after the deadline D. In the last case, the production is shortened by its entire length, obtaining less possible quality for the request based part.

The distribution of the number of extra contents $N_{E'D}|P$, given a deadline P is also different from the same quantity computed in Section 4. In particular, if we call $T_A|P$ the distribution of the time available for producing extra contents given a poll deadline P, and $a_A(t|P)$ its p.d.f., then:

$$\Pr\{N_{E'D} = n|P\} = \int_0^D \Pr\{N_E(v) = n\}\ a_A(v|P)\ dv \tag{10}$$

$T_A|P$ can be computed as the sum of two different independent distributions: $T_A|P = T_C|P + T_N|P$. In this definition, $T_C|P$ is the time available for the extra contents production during the current episode, and $T_N|P$ the one during the following episode (please recall that we have supposed that all the time before the end of the poll of one episode can still be used to produce extra contents associated to the previous one). We have that:

$$T_C|P = \max(0, D - P - M_R - \max(0, W + M_I - P)) \tag{11}$$

In Equation (11), $\max(0, W+M_I-P)$ represents the part of the main production that might exceed the poll, which is then subtracted from $D-P-M_R$ that considers the time in the slot (P, D) that is not used by the request based content. Similarly we can define:

$$T_N|P = \max(W, P - M_I) \tag{12}$$

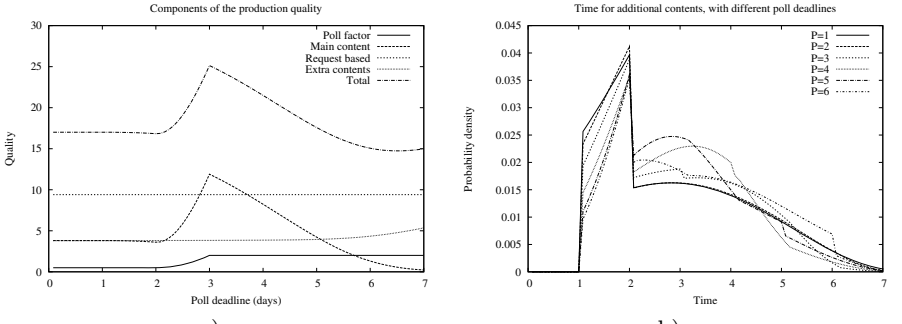
as simply the time maximum between the waiting time, and the time not used by the production of the institutional content in time slot $(0, P)$. Similar to the first case, we can define:

$$\bar{Q}_{E'}(P) = \sum_{i=0}^{N_e} \Pr\{N_{E'D}(t) = i|P\} q_e(i) \tag{13}$$

We can use the previous definitions to compute the average production quality for the institutional content case $\bar{Q}_2(P)$, given a deadline P, as:

$$\bar{Q}_2(P) = \bar{Q}_{M_I} + \bar{Q}_{M_R}(P)\bar{Q}_F(P) + \bar{Q}_{E'}(P) \tag{14}$$

Note that in this case the feedback quality modifies only the quality of the feedback dependent part of the production.



a) b)

Fig. 5. a) components of the production quality, b) distribution of the time available for producing extra

In Figure 5*a*) the various contributions to the overall quality are presented. In that case, we imagine answers to the poll arriving uniformly distributed between 2 and 3 days, and we use the same poll quality function as in the previous example. Also the waiting time was chosen uniformly distributed between 1 and 2 days. Main content and request based content production times were both considered Normal distributed, with mean 3 and variance 2, and mean 1 and variance 1 (expressed in days) respectively. We considered a maximum of 8 additional contents, each with a Normal distributed production time, characterized by mean 1 day, and variance 0.5. The poll quality function was chosen to be as

in Equation (7), with $\alpha = 1.5$, and both the institutional content and request based content quality functions where set as in Equation (3), with $q_{\max_I} = 10$ and $q_{s_I} = 1$, and $q_{\max_R} = 5$ and $q_{s_R} = 0.5$ respectively. In Figure 5*b*) the distribution of the time available for producing extras, $T_A|P$ is investigated for different values of the poll deadline P. It is interesting to see how the finite support characteristic of the waiting time distribution is clearly visible by the tooth shaped peak on the left side of the figure. As the poll deadline increases, the density of the probability mass moves on the right side of the figure, leaving thus an higher chance of producing more extra contents.

6 Parallel Production

Usually, several different units take part in the production of a single episode. We extend the model presented in Section 5 to include also this feature, as shown in Figure 6. We imagine the main content of the episode being divided into n poll based segments, and m institutional contents. The production is then carried on by k independent pipelines that works in parallel toward the common goal of completing both the main and the extra contents. The number of pipelines is not related to the number of institutional contents, neither to the number of expected request based contents, nor to the maximum number of extra contents. As in the second model, we assume that the institutional contents have priority over the request based contents. Also, each pipeline might expect a waiting time before actually starting their work. Moreover, we imagine that as soon as a troupe has finished to work on a content, it immediately starts working on another. All the produced contents are then assembled in a complete episode. We imagine that the assembly work is carried on by one of the pipelines. This model represents quite well the production of a newscast. We can imagine the pipelines as different "troupes" working on different news. To simplify the model, in this case we imagine that when the k pipelines are not working on institutional or request based contents, they can work on the extra contents. In this case we suppose that all the extra contents have to be produced before the deadline.

In this model version, M_R is divided in m smaller contents: $M_R = \sum_{j=1}^{m} M_R^j$, while M_I is divided in n smaller contents: $M_I = \sum_{j=1}^{n} M_I^j$. W, is the waiting time for that each pipeline has to wait before being operative on the episode. After all the main contents, institutional or request based, are completed, the assembly and publishing phase is modelled by the random variable A.

To simplify the analysis, we suppose that all the activities durations are characterized by exponential distributions. We implement the case proposed in Figure 6 with the Petri Net presented in Figure 7. Here white boxes represent exponentially distributed timed transition, gray boxes deterministic timed transitions, while narrow black boxes immediate transitions. All timed transitions have infinite server semantic, and their mean firing time is written below them.

Place p_1 contains as many token as production pipelines (i.e. k). Each production unit becomes active only after W: this fact is modelled by transition T_1. Place p_2 contains the pipeline that are operative: that is production units that

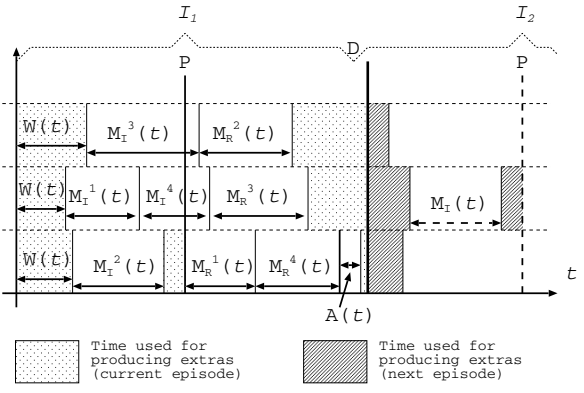


Fig. 6. Parallel production

can actually devote their effort to the realization of the considered episode. The number of institutional contents m that composes the production is represented by the initial marking of place p_3. In the same way, the initial marking n of place p_4 counts the request based contents required for this episode. Immediate transitions t_1 and t_2 allocates available production pipelines to either type of contents. Institutional contents have priority over poll based ones. This is modelled by the inhibitor arc that connects place p_3 to transition t_2, allowing it to fire only when all the institutional contents have started their production. Deterministic transition T_7 represents the end of the poll. As soon as the poll period is over, this transition moves the token from place p_9 to p_{10}. An input/output ('test') arc connecting place p_{10} to transition p_2 prevents request based contents to start their production before the end of the poll. Places p_5 and p_6 represent respectively the number of institutional and request based content currently in production. The actual institutional content production is modelled by exponential transition T_2 with mean firing time M_I, while T_3 of mean firing time M_R models the production of request based contents. Place p_7 holds the content produced. As soon as all the $n + m$ contents have been produced, immediate transition t_3 becomes enabled, and starts the assembly phase by putting a token in place p_8. Assembly is modelled by transition T_4 with mean firing time A. If assembly is completed, a token is placed in place p_{14}. The production of extra contents is modelled by the subnet on the right hand side of Figure 7. In particular, place p_{12} holds the maximum number N_e of extra contents that might be generated. The generation is modelled by transitions T_5 and T_6, both with mean firing time E. The two transitions represent whether the resources used to produce this extra has already become active for main contents (T_5), or not (T_6). Place p_{13} counts the number of extra contents actually produced. The end of the deadline is modelled by Deterministic transition T_8, characterized by firing time D, and enabled by the token in place p_{11}.

The proposed model is a Generalized Stochastic Petri Net (GSPN [5]), with two deterministic transition. However it has the properties that the poll end

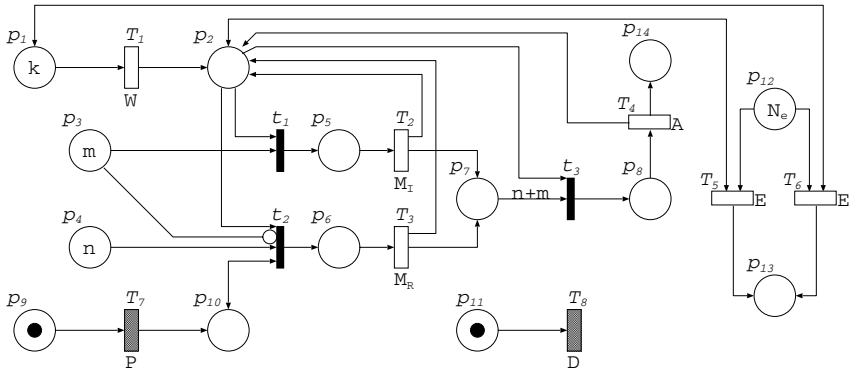


Fig. 7. Petri Net model for the multiple institutional and request based contents case

$P < D$ always happen before the deadline, and that both deterministic transitions T_7 and T_8 becomes enabled at the same time. For these reasons it can analyzed by considering two different GSPNs: one where place p_9 is marked, and another where place p_{10} is marked. Transient analysis of the first GSPNs is carried on from initial marking up to time P. Then the obtained state probability distribution is mapped to the initial state of the second GSPNs. Transient analysis is then applied to consider the evolution for the remaining $D - P$ time. Let us call $\pi(\mathcal{S}, P)$ the probability of reaching a state with the marking configuration $\mathcal{S}$ at the end of the deadline, given P. Let us assume that the total quality is linear in the number of productions, with coefficients Q_I, Q_R and Q_E for the number of institutional contents, poll based contents and extra contents respectively. Let us also assume that there is an extra quality gain Q_A when the system can conclude also the assembly phase. We can then compute the mean production quality $\bar{Q}_3(P)$ as:

$$\begin{aligned}\bar{Q}_3(P) = &\sum_{i=1}^{m} i \cdot Q_I \cdot \pi(m - (\#p_3 + \#p_5) = i, P) + \\ &+ \sum_{i=1}^{n} i \cdot Q_R \cdot \pi(n - (\#p_4 + \#p_6) = i, P) \cdot \bar{Q}_F(P) + \\ &+ \sum_{i=1}^{N_e} i \cdot Q_E \cdot \pi(\#p_{13} = i, P) + Q_A \cdot \pi(\#p_{14} = 1, P)\end{aligned}$$

where $\#p_i$ represent the number of tokens in place p_i. Note that both the number of institutional and request based contents produced are computed by subtracting to the total number of required contents (m or n), the number of contents still waiting to be produced (places p_3 and p_4) or currently in production but not yet finished (places p_5 and p_6). Note also, that as in Section 5, the quality of the poll ($\bar{Q}_F(P)$) influences only the request based contents.

In Figure 8 the quality components composing this model are shown. For this example we assumed a production having a deadline of 12 hours. We imagined

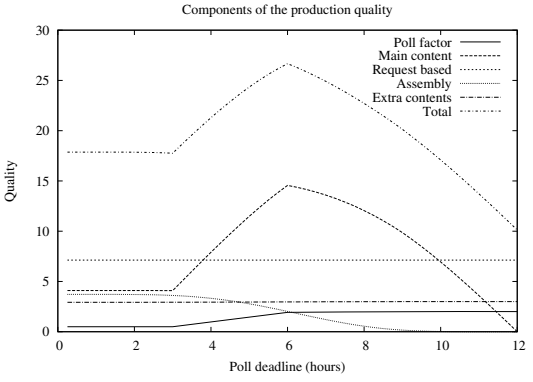


Fig. 8. The quality components

3 different working pipelines working on 6 institutional contents, 6 poll based contents and just 2 extra contents, to be assembled in 1 hour. The quality coefficient where set to: $Q_I = 1.5$, $Q_R = 2$, $Q_E = 1.5$ and $Q_A = 6$. The poll quality was considered as in previous models, but feedback was expected to arrive as a mixture of two uniform distribution, one between 3 and 6 hours with probability .95, and the other between 6 and 9 hours with probability .05.

It is easy to notice how both the extra contents ant the institutional contents are always finished before the deadline. i.e. their contribution is not almost unaffected by the time at which the poll is set. The contribution given by completing the assembly of the different parts in time is, instead, doomed to decay the more we approach the deadline. The more poll quality we have, the more overall quality we can obtain, but approaching the deadline can only decrease this quality, since there is a smaller chance to finish the contents in time.

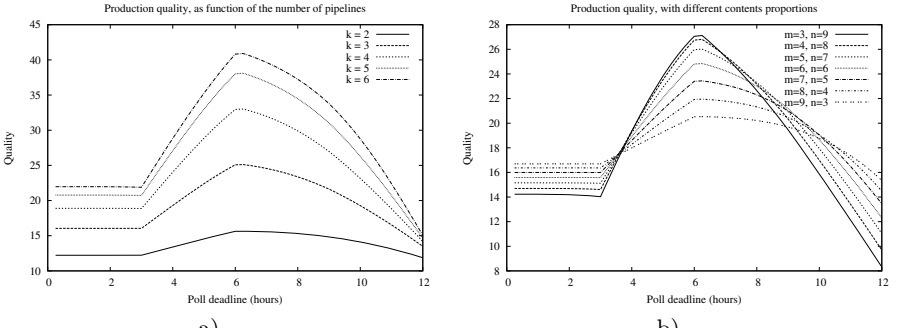


Fig. 9. Quality as function of the a) number of production components, b) proportion of institutional and request based contents

In Figure 9, we studied the overall quality varying different system configuration: in a) the number k of pipelines working to the production and in b) the ratio between institutional and poll based main contents. The case a) shows that the maximum quality level obtainable is proportional to the number of pipelines available. However, as we approach the case where each content (institutional, request based and extra) has a pipeline (that would be $k = 14$), the gain in increasing the number of the troupes becomes more limited. Case b) shows how we can obtain higher quality by increasing the number of the poll based contents. However, if we do not allow sufficient time for the production to complete the request based contents, the total quality degrades much faster, as shown by the higher slope of the curves with a higher number of request based contents.

7 Conclusions

In this paper we have studied three models, of increasing complexity, of multimedia productions based on users feedbacks. In particular we have focused our studies on the optimization of the poll deadline, by computing all the related performance indices as function of P. However, the same models, might be used in larger studies, and optimized against other parameters, such as for example the deadline D, or the quality requirements. This work should be considered as a preliminary work on a topic largely not yet explored that aims to the application of existing methodologies of stochastic performance evaluation to multi channels productions. Thus we are still acquiring the experimental data so important for the complete validation of the proposed models. The publication of these data is intended to be the goal of one of our next papers.

Acknowledgments

This work was supported by the Italian project PRIN No. 2007J4SKYP 003.

References

1. BBC future TV, `www.bbc.co.uk/futuretv`
2. RaiClick portal, `http://www.raiclick.rai.it`
3. Shear Madness web site, `http://www.shearmadness.com`
4. ICM online video survey for bbc news (2006), `http://www.icmresearch.co.uk/pdfs/2006_november_bbc_online_mobile_video_telephone.pdf`
5. Ajmone Marsan, M., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G.: Modelling with Generalized Stochastic Petri Nets. John Wiley & Sons, Chichester (1995)
6. Bettega, S.M., Fioravanti, F., Gigli, L., Grassi, G., Spinu, M.B.: Automated solutions for cross media content and multi-channel distribution. In: Proc. of AXMEDIS 2008, pp. 57–62 (2008)
7. Czyrnek, M., Kusmierek, E., Mazurek, C., Stroinski, M.: New services for iTVP content providers to manage live and on-demand content streaming. In: Proc. of AXMEDIS 2008, pp. 180–186 (2008)

8. Gawlinski, M.: Interactive Television Production. Focal Press (2003)
9. Göbel, S., Salvatore, L., Konrad, R.: StoryTec: A digital storytelling platform for the authoring and experiencing of interactive and non-linear stories. In: Proc. of AXMEDIS 2008, pp. 103–110 (2008)
10. Pemberton, L., Masthoff, J.: Adaptive hypermedia for personalised television, pp. 246–263. IRM Press (2005)
11. Pfleiderer, R., Kullen, R.: Eiaa cross media research study 2003. White paper, EIAA (2003)
12. Reitz, J.M.: Online Dictionary for Library and Information Science. Libraries Unlimited (2006)

Modeling and Analysis of Checkpoint I/O Operations

Sarala Arunagiri[1], John T. Daly[2], and Patricia J. Teller[1]

[1] The University of Texas at El Paso
{sarunagiri,pteller}@utep.edu
[2] The Center for Exceptional Computing
john.t.daly@ugov.gov

Abstract. The large scale of current and next-generation massively parallel processing (MPP) systems presents significant challenges related to fault tolerance. For applications that perform periodic checkpointing, the choice of the checkpoint interval, the period between checkpoints, can have a significant impact on the execution time of the application and the number of checkpoint I/O operations performed by the application. These two metrics determine the frequency of checkpoint I/O operations performed by the application and, thereby, the contribution of the checkpoint operations to the demand made by the application on the I/O bandwidth of the computing system. Finding the optimal checkpoint interval that minimizes the wall clock execution time has been a subject of research over the last decade. In this paper, we present a simple, elegant, and accurate analytical model of a complementary performance metric - the aggregate number of checkpoint I/O operations. We present an analytical model of the expected number of checkpoint I/O operations and simulation studies that validate the analytical model. Insights provided by a mathematical analysis of this model, combined with existing models for wall clock execution time, facilitate application programmers in making a well informed choice of checkpoint interval that represents an appropriate trade off between execution time and number of checkpoint I/O operations. We illustrate the existence of such propitious checkpoint intervals using parameters of four MPP systems, SNL's Red Storm, ORNL's Jaguar, LLNL's Blue Gene/L (BG/L), and a theoretical Petaflop system.

1 Introduction

As Massively Parallel Processing (MPP) systems scale to tens of thousands of nodes, reliability and availability become increasingly critical. Scientists have predicted that three of the most difficult and growing problems in future high-performance computing (HPC) installations will be - avoiding, coping with, and recovering from failures. With the increase in the scale of computing systems, element failures become frequent, making it increasingly difficult for long running applications to make forward progress in the absence of fault tolerance mechanisms [5].

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 386–400, 2009.

Checkpoint restart is a common technique to provide fault tolerance for applications running on MPP systems. Checkpointing can be either application-directed or system-directed. An application's *checkpoint data* is data that represents a consistent state of the application that can be saved and then, in the event of a failure, restored and used to resume execution at the saved state. A checkpoint is generally stored to persistent media (e.g., a file system). *Checkpoint latency* is the amount of time required to write checkpoint data to persistent storage and a *checkpoint interval* is the application execution time between two consecutive checkpoint operations. *Checkpoint overhead* is the increase in the execution time of an application due to checkpointing.

In a disk-based periodic checkpointing system, selecting an appropriate checkpoint interval is important especially since the storage system is physically separated from the processors used for execution of the scientific application. If the checkpoint interval is too small, the overhead created by network and storage transfers of a large number of checkpoints can have a significant impact on performance, especially when other checkpointing applications share the network and storage resources. Conversely, if the checkpoint interval is too large, the amount of work lost in the event of a failure can significantly increase the time to solution. Deciding upon the optimal checkpoint interval is the well known *optimal checkpoint interval* problem. Most solutions attempt to minimize total execution time (i.e., the application time plus the checkpoint overhead) [18] [3] [15]. In this paper we focus on another performance metric, the number of checkpoint I/O operations performed during an application run.

1.1 Motivation

The rate of growth of disk-drive performance, both in terms of I/O operations per second and sustained bandwidth, is smaller than the rate of growth of the performance of other components of computing systems [15]. Therefore, in order to attain good overall performance of computing systems, it is important to design applications to use the I/O resources efficiently, bearing in mind the limitations posed by them. There are several scientific papers that elaborate on this problem, an example of a recent paper is [15].

I/O operations performed by an application can be segregated into productive I/O and defensive I/O. Productive I/O is the component that is performed for actual science such as visualization dumps, whereas defensive I/O is the component used by fault tolerance mechanisms such as checkpoint/restart. In large applications, it has been observed that about 75% of the overall I/O is defensive I/O [1]. As indicated by [5] and other scientific literature, the demand made by checkpoint (defensive) I/O is a primary driver of the sustainable bandwidth of high performance filesystems. Hence, it is critical to manage the amount and rate of defensive I/O performed by an application. In a recent paper [15] extensive results are presented showing that as the memory capacity of the system increases so does the I/O bandwidth required to perform checkpoint operations at the optimal checkpoint interval that attains the minimum execution time. An example presented in the paper is for a system with an MTBF of 8 hours and

memory capacity of 75TB. When the checkpoint overhead is constrained to be less than or equal to 20% of application solution time, there is no solution for the optimal checkpoint interval unless the I/O bandwidth is larger than 29GB/sec. They define *utility in a cycle* as the ratio of time spent doing useful calculations to the overall time spent in a cycle and show that the I/O bandwidth required to achieve a utility of 90% is higher than what is available for present systems. Thus, while performing checkpoints at the optimal checkpoint interval that minimizes execution time, if we either restrict the checkpoint overhead to less than or equal to 20% of solution time or expect a utility greater than or equal to 90%, the I/O bandwidth required is often larger than what is available at present.

Our efforts are focused towards enhancing an understanding of the variation of the volume of generated defensive I/O, as a function of the checkpoint interval. The contributions of this paper are:

- In Section 3, we present a simple and elegant analytical model of the aggregate number of checkpoint I/O operations and a mathematical analysis of its properties that have a bearing on system performance.
- In Section 4, we present results of Monte Carlo simulations that were performed to validate the analytical model. The results show that
  - The model is accurate by demonstrating that it has a small relative error.
  - The idealization used in our analytical modeling is reasonable and it does not introduce large errors.
- In Section 5 we discuss the performance implications inferred from the mathematical analysis of Section 3.
- In Section 6, based on Poisson Execution Time Model, described next, and the modeling studies presented in this paper, we show the existence of propitious checkpoint intervals using parameters of four MPP systems, Red Storm, Jaguar, BlueGene/L, and the Petaflop machine.

Finally, in Sections 7 and 8 we present related work and future work, respectively.

2 The Poisson Execution Time Model (PETM)

The work presented in this paper is based on and complementary to the following execution time model formulated by John Daly. The total wall clock time to complete the execution of an application, the optimal checkpoint interval, and an approximate optimal checkpoint interval are given by:

$$T = Me^{R/M}\left(e^{(\tau+\delta)/M} - 1\right)\frac{T_s}{\tau} \quad \text{for} \quad \delta << T_s$$
$$\tau_{\text{opt}} = M\left(1 + ProductLog\left(-e^{-\frac{\delta+M}{M}}\right)\right)$$

The approximation to τ_{opt}, τ_{appx}, is given by

$$\tau_{\text{appx}} = \sqrt{2\delta M}\left[1 + \frac{1}{3}\left(\frac{\delta}{2M}\right)^{\frac{1}{2}} + \frac{1}{9}\left(\frac{\delta}{2M}\right)\right] - \delta \quad \text{for} \quad \delta < 2M$$
$$= \text{M} \quad \text{for } \delta \geq 2M$$

where

T_s = application solution time,

τ = checkpoint interval,
δ = checkpoint latency,
M = mean time between interruptions (MTTI) of the application, and
R = restart time.

In this paper, for the sake of convenience, we refer to the execution time model and the model of the optimal checkpoint interval presented above as the Poisson Execution Time Model (PETM) and the ProductLog Optimal Checkpoint Interval Model w.r.t Execution time (POCIME), respectively. Note that in the original literature [3], which presents these models, the terms PETM and POICME are not used to refer to the models. We introduce these terms with permission from the author of that literature.

3 Modeling the Number of Checkpoint I/O Operations: ProductLog Optimal Checkpoint Interval Model w.r.t I/O(POCIMI)

The set of I/O operations performed by a checkpoint/restart mechanism is comprised of reads and writes. In a periodic checkpointing system we know that checkpoint writes are performed periodically at every checkpoint interval and, therefore, the number of checkpoint write operations is given by the solution time of the application divided by the checkpoint interval.

$$\text{Expected number of checkpoint writes} = T_{\text{s}}/\tau$$

When a failure occurs in a periodic checkpointing system, the last checkpoint data that was successfully written needs to be read to restart the application. Therefore, the number of checkpoint read operations is given by the expected number of failures.

$$\text{Expected number of checkpoint reads} =$$
$$\frac{\text{Expected execution time}}{M} = \frac{T_{\text{s}} e^{R/M}(e^{\frac{\delta+\tau}{M}}-1)}{\tau}.$$

Expected number of aggregate checkpoint I/O operations,

$$N_{\text{I/O}} = \frac{T_{\text{s}}}{\tau}\left[1 + e^{R/M}\left(e^{\frac{\delta+\tau}{M}} - 1\right)\right] \tag{1}$$

For values of parameters MTTI = 24 hours, checkpoint latency = 5 minutes, restart time = 10 minutes, and solution time = 500 hours, using the expression for the number of checkpoint I/O operations from POCIMI and the expression for execution time from PETM, we obtain the plot shown in Fig. 1. From modeling studies in [3], we know that the execution time is a convex function of the checkpoint interval and it has a single minimum at $\tau_{\text{opt}} = 117$ minutes. From Fig. 1, it appears like $N_{\text{I/O}}$, the aggregate number of checkpoint I/O operations,

also is a convex function of the checkpoint interval, with a minimum value in the range $0 \leq \tau \leq M$. In this case, the minimum is 1,436 minutes, which is larger than the value of τ_{opt}, 117 minutes. It is important to know if these properties are invariant with respect to parameter values. In the rest of this section we present mathematical proof that the properties observed are, indeed, true for any given set of parameters.

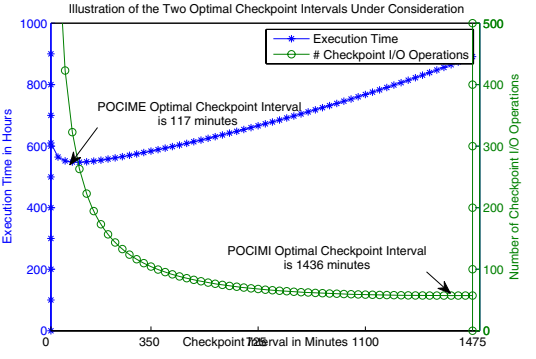


Fig. 1. Plots of Execution Time and the Number of Checkpoint I/O Operations as functions of checkpoint Interval. The parameters are MTTI, M = 24 hours, Checkpoint latency, δ = 5 minutes, and Restart time, R = 10 minutes, and Solution time, T_{s}=500hrs.

Theorem 1. *The function $N_{\text{I/O}}$ has a single minimum in the range $0 \leq \tau \leq M$; let us denote it by $\tau_{\text{I/O}}$. $N_{\text{I/O}}$ does not have any other stationary points in this range. $\tau_{\text{I/O}}$ is given by*

$$\tau_{\text{I/O}} = M\left(1 + ProductLog\left(-e^{-\frac{\delta+M}{M}} + e^{-\frac{R+\delta+M}{M}}\right)\right) \tag{2}$$

Proof. $N_{\text{I/O}}$ is given by Equation 1. We look for stationary points of $N_{\text{I/O}}$ w.r.t. τ , i.e., values of τ at which the first derivative of $N_{\text{I/O}}$ w.r.t τ is zero.

$$\frac{dN_{\text{I/O}}}{d\tau} = \frac{T_{\text{s}}}{\tau^2}\left[\frac{\tau}{M}e^{\frac{R}{M}}e^{\frac{\delta+\tau}{M}} - \left(e^{\frac{R}{M}}(e^{\frac{\delta+\tau}{M}} - 1)\right) - 1\right]$$

$$\left(\frac{dN_{\text{I/O}}}{d\tau} = 0\right) \Longrightarrow \left(e^{\frac{R}{M}}e^{\frac{\delta+\tau}{M}}\frac{\tau}{M} - e^{\frac{R}{M}}e^{\frac{\delta+\tau}{M}} + e^{\frac{R}{M}} - 1 = 0\right) \Longrightarrow$$

$$\left(e^{\frac{R}{M}}e^{\frac{\delta+\tau}{M}}\left(\frac{\tau}{M} - 1\right) = 1 - e^{\frac{R}{M}}\right) \Longrightarrow \left(e^{\frac{\delta+\tau}{M}}\left(\frac{\tau}{M} - 1\right) = -\left(1 - e^{-\frac{R}{M}}\right)\right) \Longrightarrow$$

$$\left(e^{\frac{\tau}{M}}\left(\frac{\tau}{M} - 1\right) = -e^{-\frac{\delta}{M}}\left(1 - e^{-\frac{R}{M}}\right)\right) \Longrightarrow \left(e^{\left(\frac{\tau}{M}-1\right)}\left(\frac{\tau}{M} - 1\right) = -e^{-\frac{\delta+M}{M}}\left(1 - e^{-\frac{R}{M}}\right)\right) \Longrightarrow$$

$$\left(\left(\frac{\tau}{M} - 1\right) = ProductLog\left(-e^{-\frac{\delta+M}{M}}\left(1 - e^{-\frac{R}{M}}\right)\right)\right) \Longrightarrow$$

$$\left(\frac{\tau}{M} = 1 + ProductLog\left(-e^{-\frac{\delta+M}{M}}\left(1 - e^{-\frac{R}{M}}\right)\right)\right) \Longrightarrow$$

$$\left(\tau = M\left(1 + ProductLog\left(-e^{-\frac{\delta+M}{M}}\left(1 - e^{-\frac{R}{M}}\right)\right)\right)\right)$$

$$\tau = M\left(1 + ProductLog\left(-e^{-\frac{\delta+M}{M}} + e^{-\frac{R+\delta+M}{M}}\right)\right) \tag{3}$$

There is a unique positive value of τ that satisfies the above equation; let us denote it by $\tau_{\mathrm{I/O}}$. The *ProductLog* term in Equation 3 is negative and its absolute value is less than one. Therefore, $\tau_{\mathrm{I/O}}$ is always less than M. We use the second derivative test in order to determine whether the stationary point $\tau_{I/0}$ is a minimum, maximum, or an inflexion point.

We know that

$$\begin{aligned} N_{\mathrm{I/O}} &= \frac{\text{Expected Execution Time}}{M} + \frac{T_{\mathrm{s}}}{\tau} = \frac{T}{M} + \frac{T_{\mathrm{s}}}{\tau} \\ \frac{dN_{\mathrm{I/O}}}{d\tau} &= \frac{1}{M}\frac{dT}{d\tau} - \frac{T_{\mathrm{s}}}{\tau^2} \\ \frac{d^2N_{\mathrm{I/O}}}{d\tau^2} &= \frac{1}{M}\frac{d^2T}{d\tau^2} + 2\frac{T_{\mathrm{s}}}{\tau^3} \end{aligned} \tag{4}$$

From [3] we know that $\frac{d^2T}{d\tau^2}$ is positive for all values of τ in the range $0 < \tau \leq M$. This makes the right-hand side of Equation 4 and, thus, $\frac{d^2N_{\mathrm{I/O}}}{d\tau^2}$ positive for all τ in the range $0 < \tau \leq M$. Therefore, the stationary point $\tau_{\mathrm{I/O}}$ is a minimum with respect to the number of I/O operations. □

We now investigate the relationship between $\tau_{\mathrm{I/O}}$ and τ_{opt} for any given set of checkpoint parameters.

Theorem 2. *The value of the checkpoint interval that minimizes the number of I/O operations, $\tau_{\mathrm{I/O}}$, is always greater than the value of the checkpoint interval that minimizes the expected execution time, τ_{opt}.*

Proof. Recall the expressions for τ_{opt} and $\tau_{\mathrm{I/O}}$;

$$\begin{aligned} \tau_{\mathrm{opt}} &= M\left(1 + ProductLog\left(-e^{-\frac{\delta+M}{M}}\right)\right) \\ \tau_{\mathrm{I/O}} &= M\left(1 + ProductLog\left(-e^{-\frac{\delta+M}{M}} + e^{-\frac{R+\delta+M}{M}}\right)\right) \end{aligned}$$

Consider arguments to the *ProductLog* function in the above equations for τ_{opt} and $\tau_{\mathrm{I/O}}$. They are both negative and the absolute value of the argument in the equation for τ_{opt} is larger than that of the equation for $\tau_{\mathrm{I/O}}$. Since $ProductLog(-1/e) = -1$ and the *ProductLog* function is monotonically increasing in the range ($-\frac{1}{e}$ to 0).

$$\begin{aligned} &|ProductLog\left(-e^{-\frac{\delta+M}{M}}\right)| > |ProductLog\left(-e^{-\frac{\delta+M}{M}} + e^{-\frac{R+\delta+M}{M}}\right)| \\ &\Longrightarrow \tau_{\mathrm{opt}} < \tau_{\mathrm{I/O}} \end{aligned}$$

□

Thus, as illustrated by Fig. 1, we have established that for checkpoint intervals τ in the range $\tau_{\mathrm{opt}} \leq \tau \leq \tau_{\mathrm{I/O}}$, the number of checkpoint I/O operations decreases with increasing checkpoint intervals.

Corollary 1. *For checkpoint intervals, τ, in the range $\tau_{\text{opt}} \leq \tau \leq \tau_{\text{I/O}}$, the expected value of the frequency of checkpoint I/O operations decreases as the checkpoint interval increases.*

Proof. We know from PETM that for values of checkpoint intervals, τ, in the range $\tau_{\text{opt}} \leq \tau \leq M$, the expected execution time increases as the checkpoint interval increases. Since $\tau_{\text{I/O}} < M$, it follows that the expected execution time increases as the checkpoint interval increases for τ in the range $\tau_{\text{opt}} \leq \tau \leq \tau_{\text{I/O}}$. This information and Theorem 2 together imply that for checkpoint intervals, τ, in the range $\tau_{\text{opt}} \leq \tau \leq \tau_{\text{I/O}}$, the expected value of the frequency of checkpoint I/O operations decreases as the checkpoint interval increases. □

In order to evaluate the accuracy of our analytical model, POCIMI, it is infeasible, in terms of system availability, execution time, and effort, to conduct repeated runs of experiments on the scale of systems that we are studying. Thus, the only feasible alternative for us is a simulation study, which we describe and discuss next.

4 Monte Carlo Simulation to Validate the Analytical Model, POCIMI

The goal of our simulation study was to validate the accuracy of the analytical model for the number of checkpoint I/O operations, POCIMI, by comparing the numbers estimated by POCIMI with those obtained using simulation of the execution of an application on an MPP system.

4.1 Details of Simulation

The simulator was coded using MATLAB to perform a discrete event simulation of the physical process of running an application on a 1,000-node system with each node having an exponential failure distribution. The events in the simulation were confined to those relevant to the process of checkpoint/restart. Failure times were generated using random number generators and, as time progresses, the number of checkpoint reads, number of checkpoint writes, execution time, and number of failures are counted until the application completes execution.

Six sets of simulations were performed, one for each of the following values of checkpoint latency: 5,10,15,20,25, and 30 minutes. The other parameter values were set as follows: solution time of the simulated application: 500 hours, restart time: 10 minutes, and MTTI of the parallel system: 24 hours or 1440 minutes. These parameter values were picked from examples in the published literature. Each set of experiments had five trials. The design variable was the checkpoint interval and the response variable was the number of checkpoint I/O operations. During each trial, the values of the response variable, i.e., the number of checkpoint I/O operations, were counted; each value corresponds to a different value of the design variable, i.e., the checkpoint interval. The range of interest for values of the checkpoint interval was 0 to 1440. We split this range into

three subintervals, low values, medium values, and high values, and picked six data points within each subinterval. Accordingly, the design points of our simulation study were the following 18 values of checkpoint intervals: {50,75,...,175, 650,675,...,775,1350,1375,...,1475}. For each trial, and at each chosen checkpoint interval, we simulated 100 runs of the application and recorded the number of checkpoint I/O operations, in addition to other data, such as execution time and number of failures. For each trial, we calculated the average values of the metrics of interest as an arithmetic mean over the 100 runs of the trial. For the plots presented in Fig. 2, we arbitrarily picked data from one trial, i.e., Trial 3, which has a checkpoint latency of 5 minutes. The decision to depict data from only one trial was made for the sake of clarity – the lines representing the simulated mean values of all trials were almost overlapping and cluttering the figure. Subplot(a) of Fig. 2 is a plot of 99% confidence interval of the mean simulated number of checkpoint I/O operations and the number estimated by the analytical model, POCIMI. For completeness sake, we present in Subplot(b) and Subplot(c)the execution time and inter-arrival times of checkpoint I/O operations, respectively. As can be seen from the plot, at the scale at which the figure is presented, the line representing the analytical model and the one representing the simulated mean almost overlap, and the 99% confidence interval is very small. When we did zoom into the figure, we were able to see that there was, indeed, an error bar showing the confidence interval. While the plots in Fig. 2 present the trends for checkpoint intervals varying over the whole range of interest, Figs. 3 and 4 show the details. Note that unlike Fig. 2, Figs. 3 and 4 use data from all trials belonging to all sets of experiments, i.e., 30 trials in total. Subplot(a) and Subplot(b) of Fig. 3 show bar graphs that represent the range of values of absolute errors and relative errors of the 30 trials. The absolute error and relative error are defined by,

$$\text{Absolute error} = \begin{array}{c}\#\text{ checkpoint I/O operations of POCIMI} \\ \text{-mean simulated }\#\text{ checkpoint I/O operations}\end{array}$$

$$\text{Relative error} = \frac{\text{Absolute error}}{\#\text{ checkpoint I/O operations of POCIMI}} * 100$$

For the simulated number of checkpoint I/O operations for all 30 trials, Subplot(a) of Fig. 4 presents the maximum value of the size of the 99% confidence interval.

4.2 Discussion of Results

- The value of the relative error of the estimates provided by the analytical model for all 30 trials lies within ±6%. This demonstrates the degree of accuracy of the model.
- The size of the 99% confidence interval of the number of simulated checkpoint I/O operations is no more than 8% of its mean value. This implies that the aggregate number of checkpoint I/O operations from the simulation runs has a small variance.

4.3 Addressing the Idealization in Analytical Modeling

Idealization is the process by which scientific models assume facts about the phenomenon being modeled that may not be entirely accurate. Often these assumptions are used to make models easier to understand or solve. One of the caveats of analytical modeling is the idealization used in order to make the model tractable or solvable, or mathematically elegant. With an intent to quantify the contribution of idealization to the error in the predictive accuracy of POCIMI, we performed the following experiment. Corresponding to every simulated run of the application at each chosen design point, i.e., value of checkpoint interval, we ran three versions of the simulation: the base version, the idealized version, and the minimally idealized version.

The details of this experiment are presented in [2]. Subplot(b) of Fig. 4 shows the difference in relative error between the idealized version of the simulation and the minimally idealized version of the simulation. We find that the contribution to the relative error made by the idealization used in our analytical model is within the range ±2%. This demonstrates that the idealization used in POCIMI is not too restrictive and, therefore, does not affect the accuracy of the model too much.

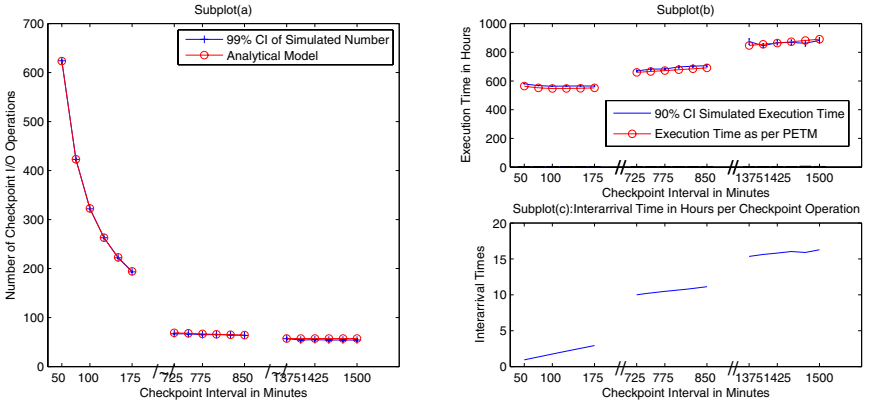


Fig. 2. Subplot(a): Number of checkpoint I/O operations as a function of the checkpoint interval. Subplot(b): Execution time versus checkpoint intervals. Subplot(c): Mean interarrival time, in hours, of checkpoint operations.

5 Performance Implications Inferred by Analyzing POCIMI

1. An insight provided by the model is that while τ_{opt} and $\tau_{\text{I/O}}$ are both functions of δ and M, $\tau_{\text{I/O}}$ is also a function of the restart time, R. $\tau_{\text{I/O}}$ decreases with increasing values of R.
2. Corollary 1 is key to promising avenues in performance improvement. For values of τ in the range $\tau_{\text{opt}} \leq \tau \leq \tau_{\text{I/O}}$, both the expected values of the frequency of checkpoint I/O operations and the number of checkpoint I/O operations decrease with increases in the checkpoint interval.

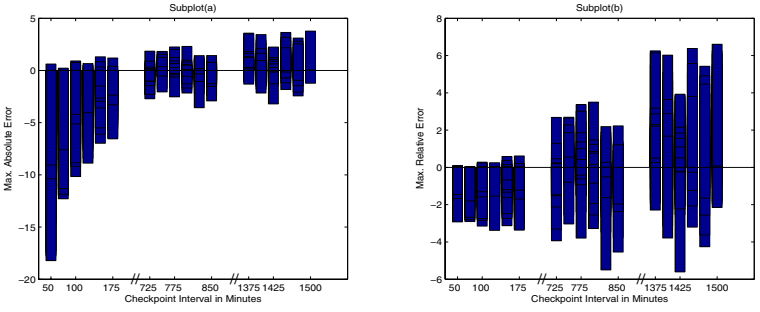


Fig. 3. Subplot(a): Absolute error of POCIMI. Subplot(b): Relative error.

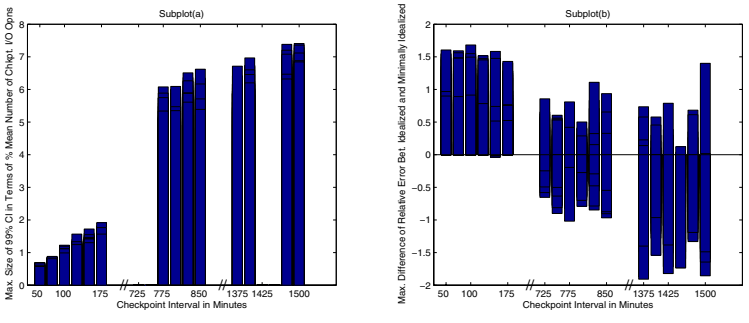


Fig. 4. Subplot(a): Size of the 99% confidence interval(CI) of the simulated number of checkpoint I/O operations. Subplot(b): Difference between relative errors of POCIMI w.r.t. the idealized and minimally idealized versions of the simulation.

3. For time-critical applications for which having a minimum wall clock execution time is important, using τ_{opt} as a checkpoint interval makes perfect sense. However, for all other applications it would be of interest to find out whether it is possible to choose a checkpoint interval that is larger than the τ_{opt} such that the corresponding execution time is marginally larger than the minimum execution time, while the corresponding number of checkpoint I/O operations is drastically smaller than its value at τ_{opt}.
4. If we explore clues from visual inspection of Fig. 1, we observe that for checkpoint intervals greater than and in the vicinity of τ_{opt}, the execution time curve rises slowly, while the curve of the number of checkpoint I/O operations falls steeply. This seems to indicate that, probably, in this region, there are checkpoint intervals such that the corresponding numbers of checkpoint I/O operations are drastically smaller than values corresponding to τ_{opt}, while the execution times are marginally larger than the minimum execution time. Whether or not this observation holds good, in general, for all values of parameters, is not clear. To know this requires a rigorous math-

ematical analysis involving gradients of the execution time function and the number of checkpoint I/O operations function, in the region of interest. This appears to be a non-trivial mathematical exercise and it could be prospective future work. Nonetheless, in the next section, we investigate this idea for specific cases using parameters from four MPP systems.

6 Investigation of Performance Improvement

In this section, using POCIME and POCIMI, we model the performance of four MPP architectures: SNL's Red Storm, ORNL's Jaguar, LLNL's Blue Gene/L (BG/L), and a theoretical Petaflop system. The values of parameters of these systems are presented in Table 1. For all experiments, we consider a representative application with a solution time, $T_s, of 500\ hours$ and a restart time, $R, of 10\ minutes$. For each of the four computing systems and the representative application, assume that the checkpoint interval is larger than τ_{opt} and the corresponding expected execution time is 105% of the minimum execution time, E_{min}, given by PETM, represented as $\tau_{1.05\text{E}_{\text{min}}}$. For the representative application running on the four MPP systems, we investigate the extent to which the number of checkpoint I/O operations corresponding to the checkpoint interval $\tau_{1.05\text{E}_{\text{min}}}$ is reduced, as compared to the number of checkpoint I/O operations at τ_{opt}. For each MPP system, assume that

- the application runs on all nodes of the system,
- the MTTI of each node is 5 years, and
- the application checkpoints half of each processor's memory at each checkpoint.

This set of assumptions is labeled *Standard*. We then consider three other variations of the standard assumptions. The first variation assumes that the application checkpoints 25% of its memory, instead of 50%. The second variation assumes that the MTTI of each node is 2.5 years, instead of 5 years. Finally, the third variation assumes that the application runs on 1/8th of the nodes of each system, instead of all the nodes. In this last case, while computing the checkpoint latency, the partition is considered to have 1/8th of the storage bandwidth available to it. These assumptions cover a few common cases.

For the sixteen cases discussed earlier, the impact of increasing the checkpoint interval, from τ_{opt} to $\tau_{1.05Emin}$, on the number of checkpoint I/O operations is

Table 1. Parameter values for the studied MPPs

Parameter	Red Storm	Blue Gene/L	Jaguar	Petaflop
$n_{max} \times cores$	$12,960 \times 2$	$65,536 \times 2$	$11,590 \times 2$	$50,000 \times 2$
d_{max}	1GB	0.25GB	2.0GB	2.5GB
M_{dev}	5 years	5 years	5 years	5 years
β_s	50GB/s	45GB/s	45GB/s	500GB/s

Table 2. Decrease in the number of checkpoint I/O operations of the representative application at $\tau_{1.05E_{min}}$

MPP System	Conditions	# checkpoint I/O operations for $\tau = \tau_{opt}$	# checkpoint I/O operations for $\tau = \tau_{1.05E_{min}}$	% decrease
Red Storm	Standard	962	587	38.94
	25% memory checkpointed	1248	669	46.35
	Partition Size: 1/8th n_{max}	***280***	***109***	***61.04***
	Node MTTI: 2.5years	1569	1091	30.48
Blue Gene/L	Standard	3407	2907	14.68
	25% memory checkpointed	3660	2773	24.24
	Partition Size: 1/8th n_{max}	***631***	***380***	***39.42***
	Node MTTI: 2.5years	6212	5571	10.25
Jaguar	Standard	712	482	32.53
	25% memory checkpointed	895	537	40.02
	Partition Size: 1/8th n_{max}	***194***	***86***	***55.63***
	Node MTTI: 2.5years	1215	924	23.94
Petaflop	Standard	2697	2100	22.35
	25% memory checkpointed	3166	2195	32.22
	Partition Size: 1/8th n_{max}	***615***	***324***	***47.22***
	Node MTTI: 2.5years	5568	4852	12.86

presented in Table 2. For each of the four systems considered, the case that has the largest decrease in the number of checkpoint I/O operations is shown in bold. The reduction in the number of checkpoint I/O operations was in the range of 10.25% to 61.07%.

7 Background and Related Work

There is a substantial body of literature regarding the optimal checkpoint problem and several models of optimal checkpoint intervals have been proposed. Young proposed a first-order model that defines the optimal checkpoint interval in terms of checkpoint overhead and mean time to interruption (MTTI). Young's model does not consider failures during checkpointing and recovery [18]. However, POCIME, which is an extension of Young's model to a higher-order approximation, does [3]. In addition to considering checkpoint overhead and MTTI, the model discussed in [16] includes sustainable I/O bandwidth as a parameter and uses Markov processes to model the optimal checkpoint interval. The model described in [11] uses useful work, i.e., computation that contributes to job completion, to measure system performance. The authors claim that Markov models are not sufficient to model useful work and propose the use of Stochastic Activity Networks (SANs) to model coordinated checkpointing for large-scale systems. Their model considers synchronization overhead, failures during checkpointing and recovery, and correlated failures. This model also defines the optimal number of processors that maximize the amount of total useful work. Vaidya models the checkpointing overhead of a uniprocess application. This model also considers failures during checkpointing and recovery [17]. To evaluate the performance and scalability of coordinated checkpointing in future large scale systems, [4] simulates checkpointing on several configurations of a hypothetical Petaflop system.

Their simulations consider the node as the unit of failure and assume that the probability of node failure is independent of its size, which is overly optimistic [6]. Yet another related area of research is failure distributions of large-scale systems. There has been a lot of research conducted in trying to determine failure distributions of systems. Failure events in large-scale commodity clusters as well as the BG/L prototype have been shown to be neither independent, identically distributed, Poisson, nor unpredictable [8] [10]. [12] presents a study on system performance in the presence of real failure distributions and concludes that Poisson failure distributions are unrealistic. Similarly, a recent study by Sahoo [14] analyzing the failure data from a large-scale cluster environment and its impact on job scheduling, reports that failures tend to be clustered around a few sets of nodes, rather than following a particular distribution. In 2004 there was a study on the impact of realistic large-scale cluster failure distributions on checkpointing [10]. Oliner et. al.[9] profess that a realistic failure model for large-scale systems should admit the possibility of critical event prediction. They also state that the idea of using event prediction for pro-active system management is a direction worth exploring [10] [13]. Recently, there has been a lot of research towards finding alternatives for disk-based periodic checkpointing techniques [9] [7] and there have been some promising results. However, until these new techniques reach a level of maturity, disk-based periodic checkpointing technique will continue to be the reliable and time-tested method of fault tolerance [15]. Besides, a lot of important legacy scientific applications use periodic checkpointing and, therefore, issues related to periodic checkpointing still need to be addressed.

Note that PETM and POCIME do not make any assumptions on the failure distribution of the system for its **entire lifetime**. However, they assume an exponential failure distribution only for the **duration of the application run**, which might be a few days, weeks, or months. Note that this is drastically different from assuming an exponential failure distribution for the life of the system. This model offers the application programmer the flexibility to use whatever means is deemed right for the system to determine the value of MTTI, M, at the beginning of the application run. Given this value of M, the model then assumes that during the application run the failure distribution of the system is exponential. This makes the model mathematically amenable, elegant, and useful. The assumption of exponential failure distribution for the duration of the application run is validated by the observation that a plot of the inter-arrival times of 2,050 single-node unscheduled interrupts, gathered on two different platforms at Los Alamos National Laboratories over a period of a year, i.e., January 2003 to December 2003, fits a Weibull distribution with a shape factor 0.91/0.97. Since an exponential distribution is equivalent to a Weibull distribution with a shape factor 1.0, it is reasonable to assume an exponential failure distribution. Due to space constraints, we do not present the plot in this paper.

8 Conclusions and Future Work

We believe that the modeling work presented in this paper, based on the POCIMI model, is complementary to that associated with the PETM and POCIME

models. Together they provide pointers and insights for making an informed tradeoff between expected execution time and the number of checkpoint I/O operations. This facilitates an application programmer to chose a value of the checkpoint interval, a tunable parameter, that balances the frequency at which the application performs checkpoint I/O operations and expected execution time. To the best of our knowledge, at this time there is no quantitative guidance to facilitate such a tradeoff. Both models do not factor in the deterioration caused by resource contention. However, they model the general case, which can be used as a guidance for specific cases.

In an MPP system that has a system-wide view of all concurrently executing applications and has control over the checkpoint parameters of these applications, checkpoint intervals could be tuned to provide performance differentiation and performance isolation of concurrent applications. For example, the application with highest priority can be run with a checkpoint interval that is optimal w.r.t execution time, while applications with the lowest priorities can be set to run with checkpoint intervals that are closer to the value of the optimal checkpoint interval w.r.t total number of checkpoint I/O operations. The other applications can, perhaps, use checkpoint intervals that are between their two optimal values. For periodic checkpointing applications, both the expected wall clock execution time and the expected number of checkpoint I/O operations are important metrics to be considered in order to make decisions about checkpoint intervals. An important target of our future work is to provide specific guidelines about how to coordinate checkpoint operations of concurrently executing applications in order to achieve high system throughput.

Acknowledgments

We are pleased to recognize the support of this work by the Army High Performance Computing Research Center (AHPCRC) under ARL grant number W11NF-07-2-2007 and the helpful professional interactions we have had with Seetharami Seelam (IBM), Ron Oldfield and Rolf Riesen (Sandia National Laboratories), and Maria Ruiz Varela (UTEP).

References

1. Asci purple statement of work, lawrence livermore national laboratory, `http://www.llnl.gov/asci/purple/attachment_02_purplesowv09.pdf` (accessed: April 23, 2006)
2. Arunagiri, S., Daly, J.T., Teller, P.J.: Propitious checkpoint intervals to improve system performance. Technical Report UTEP-CS-09-09, University of Texas at El Paso (2009)
3. Daly, J.: A higher order estimate of the optimum checkpoint interval for restart dumps. Future Generation Computer Systems 22, 303–312 (2006)
4. Elnozahy, E.N., Plank, J.S.: Checkpointing for peta-scale systems: A look into the future of practical rollback-recovery. IEEE Transactions on Dependable and Secure Computing 1(2), 97–108 (2004)

5. Gibson, G., Schroeder, B., Digney, J.: Failure tolerance in petascale computers. CTWatch Quarterly (November 2007)
6. Kavanaugh, G.P., Sanders, W.H.: Performance analysis of two time-based coordinated checkpointing protocols. In: PRFTS 1997: Proceedings of the 1997 Pacific Rim International Symposium on Fault-Tolerant Systems, Washington, DC, USA, p. 194. IEEE Computer Society, Los Alamitos (1997)
7. Kim, Y., Plank, J.S., Dongarra, J.J.: Fault tolerant matrix operations for networks of workstations using multiple checkpointing. In: HPC-ASIA 1997: Proceedings of High-Performance Computing on the Information Superhighway, HPC-Asia 1997, Washington, DC, USA, p. 460. IEEE Computer Society, Los Alamitos (1997)
8. Liang, Y., Sivasubramaniam, A., Moreira, J.: Filtering failure logs for a bluegene/l prototype. In: Proceedings of the 2005 International Conference on Dependable Systems and Networks (DSN 2005), June 2005, pp. 476–485 (2005)
9. Oliner, A.J., Rudolph, L., Sahoo, R.K.: Cooperative checkpointing: a robust approach to large-scale systems reliability. In: ICS 2006: Proceedings of the 20th Annual International Conference on Supercomputing, Cairns, Queensland, Australia, pp. 14–23. ACM Press, New York (2006)
10. Oliner, A.J., Rudolph, L., Sahoo, R.K.: Cooperative checkpointing theory. In: Proceedings of IPDPS, Intl. Parallel and Distributed Processing Symposium (2006)
11. Pattabiraman, K., Vick, C., Wood, A.: Modeling coordinated checkpointing for large-scale supercomputers. In: Proceedings of the 2005 International Conference on Dependable Systems and Networks (DSN 2005), Washington, DC, pp. 812–821. IEEE Computer Society, Los Alamitos (2005)
12. Plank, J.S., Elwasif, W.R.: Experimental assessment of workstation failures and their impact on checkpointing systems. In: Proceedings of the The Twenty-Eighth Annual International Symposium on Fault-Tolerant Computing, June 1998, pp. 48–57 (1998)
13. Sahoo, R.K., Bae, M., Vilalta, R., Moreira, J., Ma, S., Gupta, M.: Providing persistent and consistent resources through event log analysis and predictions for large-scale computing systems. In: SHAMAN Workshop, ICSY 2002 (June 2002)
14. Sahoo, R.K., Sivasubramaniam, A., Squillante, M.S., Zhang, Y.: Failure data analysis of a large-scale heterogeneous server environment. In: Proceedings of the International Conference on Dependable Systems and Networks (DSN 2004), June 2004, pp. 772–781 (2004)
15. Subramaniyan, R., Grobelny, E., Studham, S., George, A.D.: Optimization of checkpointing-related i/o for high-performance parallel and distributed computing. J. Supercomput. 46(2), 150–180 (2008)
16. Subramaniyan, R., Studham, R.S., Grobelny, E.: Optimization of checkpointing-related I/O for high-performance parallel and distributed computing. In: Proceedings of The International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 937–943 (2006)
17. Vaidya, N.H.: Impact of checkpoint latency on overhead ratio of a checkpointing scheme. IEEE Transactions on Computers 46(8), 942–947 (1997)
18. Young, J.W.: A first order approximation to the optimum checkpoint interval. Communications of the ACM 17(9), 530–531 (1974)

Author Index