Tadeusz Czachórski · Erol Gelenbe
Ricardo Lent *Editors*

Information Sciences and Systems 2014

Proceedings of the 29th International Symposium on Computer and Information Sciences

Springer

Information Sciences and Systems 2014

Tadeusz Czachórski · Erol Gelenbe
Ricardo Lent
Editors

Information Sciences and Systems 2014

Proceedings of the 29th International Symposium on Computer and Information Sciences

Springer

Editors
Tadeusz Czachórski
Polish Academy of Sciences
Gliwice
Poland

Erol Gelenbe
Ricardo Lent
Imperial College London
London
UK

ISBN 978-3-319-09464-9 ISBN 978-3-319-09465-6 (eBook)
DOI 10.1007/978-3-319-09465-6

Library of Congress Control Number: 2013947374

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The 29th Annual Symposium on Computer and Information Sciences, held in Krakow under the auspices of the Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences, pursues the tradition of a broad-based symposium presenting recent advances in scientific and technical aspects of Information Technology.

Notable in this year's event are the presence of new areas such as Quantum Computing, and the greater emphasis on Network Security via a session (as last year) that presents some of the results from the EU FP7 Project NEMESYS.

All papers included in the proceedings have undergone an evaluation by at least two referees, and most of the papers have been examined by at least three referees. Published by Springer, in the past 3 years each of these proceedings has had some 10,000 full paper downloads, which is equivalent to, on average, over 200 downloads per paper.

We therefore hope that this year's ISCIS 2014 proceedings, the 29th Symposium in a long series that started in Ankara, Turkey, in 1986, will continue to be useful to the international scientific and technical community in Computer Science and Engineering.

London, July 2014

Tadeusz Czachórski
Erol Gelenbe
Ricardo Lent

Program Committee

Ethem Alpaydin
Volkan Atalay
Cevdet Aykanat
Selim Balcisoy
Madalina Baltatu
Javier Barria
Olivier Beaumont
Leszek Borzemski
Jeremy Bradley
Manfred Broy
Wojciech Burakowski
Fazli Can
Sophie Chabridon
Ilyas Cicekli
Nihan Cicekli
Tadeusz Czachorski, IITiS PAN, Polish Academy of Sciences
Gokhan Dalkilic
Laurent Delosieres
Mariangiola Dezani
Nadia Erdogan
Engin Erzin
Taner Eskil
Jean-Michel Fourneau
Erol Gelenbe
Stephen Gilmore
Mariusz Glabowski
Krzysztof Grochla, Institute of Theoretical and Applied Informatics of PAS
Adam Grzech
Lin Guan
Ugur Gudukbay
Attila Gursoy

Ugur Halici
Peter Harrison
Yorgo Istefanopulos
Alain Jean-Marie
Sylwester Kaczmarek
Jacek Kitowski
Stefanos Kollias
Jerzy Konorski
Ibrahim Korpeoglu
Stanislaw Kozielski
Olcay Kursun
Ricardo Lent
Albert Levi
Aristidis Likas
George Limperopoulos
Peixiang Liu
Jozef Lubacz
Chris Mitchell
Marek Natkaniec
Christos Nikolaou
Sema Oktug
Ender Ozcan
Oznur Ozkasap
Zdzislaw Papir
Ferhan Pekergin
Nihal Pekergin
Michal Pioro
Yves Robert
Alexandre Romariz
Georgia Sakellari
Huseyin Seker
Ercan Solak
Andreas Stafylopatis
Maciej Stasiak
Halina Tarasiuk
Eleni Theodoropoulou
Dao Thi
Nigel Thomas
Hakki Toroslu
Salvatore Tucci
Dimitrios Tzovaras
Ozgur Ulusoy
Ozlem Uzuner
Krzysztof Walkowiak
Wei Wei

Jozef Wozniak
Zhiguang Xu
Adnan Yazici
Cemal Yilmaz
Arda Yurdakul
Thomas Zeugmann
Qi Zhu

Additional Reviewers

Bull, Peter
Cavagnino, Davide
Gelenbe, Erol
Gena, Cristina
Hanczewski, Slawomir
Liber, Arkadiusz
Quessette, Franck
Rhouma, Rhouma
Tüysüz Erman, Ayşegül
Weissenberg, Joanna
Zielinski, Bartlomiej

Contents

Part I
Wireless and Cognitive Networks

New Channel Access Approach for the IEEE 802.15.4 Devices in 2.4 GHz ISM Band

Tolga Coplu and Sema F. Oktug

Abstract The number of indoor wireless communication devices and technologies employing the ISM band is increasing day by day. As a consequence, co-existence is a big challenge for tiny, IEEE 802.15.4-based and resource-constrained devices. In this paper, 2.4 GHz ISM band aggregated traffic is analyzed and a novel channel access approach is proposed based on the cognitive radio concepts in order to increase free channel access performance. The performance is evaluated by using the real-world RF signal strength measurements of an indoor IEEE 802.15.4 node with the presence of many IEEE 802.11 interferers. The performance evaluation gives promising results considering free channel access performance.

Keywords Co-existence · Cognitive radio · IEEE 802.15.4 · IEEE 802.11 · Channel access

1 Introduction

We are facing a limited bandwidth when communicating indoor wirelessly. In the 2.4 GHz Industrial, Scientific and Medical (ISM) band, the variety and the number of devices have a huge impact on the efficient use of the available bandwidth. As a result, co-existence raises a big challenge especially for the low power devices.

There are very valuable studies on the 2.4 GHz ISM band co-existence to mitigate the interference [1–5]. These studies especially focus on the co-existence of IEEE 802.11 and IEEE 802.15.4 technologies, because of their asymmetric interference

T. Coplu (✉) · S.F. Oktug
Department of Computer Engineering, Istanbul Technical University, Istanbul, Turkey
e-mail: coplu@itu.edu.tr

S.F. Oktug
e-mail: oktug@itu.edu.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_1

patterns, as stated in [5, 6]. In all these techniques, the main principle is to decrease the conflicting resources. Since 2.4 GHz ISM band communication technologies are generally channel based, channel is one of the most important resources. Thus, many of the schemes in the literature propose dynamic or adaptive channel selection for better co-existence performance. Second, time is also a very valuable resource. In wireless media, there is no effective and fully utilized communication under co-existence. Therefore, determining free time slots has a key role for inner channel interference avoidance. Finally, space is another resource. Unfortunately, it cannot be effectively used since communicating devices do not have controlled mobility. However, there are important transmit power control schemes in the literature for interference mitigation.

In this work, we focus on boosting 2.4 GHz ISM band inner channel co-existence performance by using the cognitive radio concepts. The authors consider the IEEE 802.15.4 devices as the secondary users in the cognitive radio networks and the IEEE 802.11 devices as the primary users and introduce a new scheme for the channel access to be used by the IEEE 802.15.4 devices. The performance of the introduced scheme is studied under various real-life IEEE 802.11 traffic traces and promising results are obtained.

This paper is organized as follows: The analogy between ISM band co-existence and the cognitive radio concepts is given in Sect. 2. Section 3 presents the impacts of aggregated IEEE 802.11 traffic on the channel access performance of IEEE 802.15.4 devices. In Sect. 4, a new channel access scheme is explained in detail. Section 5 gives the performance evaluation of the proposed scheme using a testbed. Finally, Sect. 6 concludes the paper by giving future directions.

2 Motivation

In cognitive radio, the users are classified as primary and secondary users [7, 8]. Primary users have higher priority over secondary users in terms of communication due to a license mechanism. Unlike in the cognitive radio communications, there is no defined priority access mechanism in license-free ISM bands but a spontaneously occurring privileged access due to different output power of devices. For example, an IEEE 802.11 device interferes with a nearby IEEE 802.15.4 device communication and prevents its channel access or successful transmission whereas the IEEE 802.15.4 device has no such effect on the IEEE 802.11 communication [6].

At this point, it is practical to make an analogy with cognitive radio and adapt its priority-based user differentiation concept in the ISM band co-existence problem. Within the scope of this paper, the nodes suppressing the communication of the neighboring nodes are going to be called as ISM Primary Users (ISM-PU) and the aggregated heterogeneous traffic generated by these ISM-PUs is going to be called as Aggregated ISM-PU Traffic. On the other hand, the nodes affected by the aggregated ISM-PU traffic are going to be called as ISM Secondary Users (ISM-

SU). However, there are some domain specific differences between these two cases. Unlike in cognitive radio:

- The number of ISM-PUs is not restricted to one,
- ISM-PUs are not static. They may lose their ISM-PU characteristics in time because of mobility or change in Radio Frequency (RF) output power (vice versa),
- Devices using the same or different kinds of communication protocols can be the ISM-PU of the device that is of interest. For example, either an IEEE 802.11 or an IEEE 802.15.4 device can be the ISM-PU of an IEEE 802.15.4 device interested,
- Primary and secondary terms are not global. A node may not be the ISM-PU of all its neighbors.

Evaluating all the items above, it is hard to fully adapt cognitive radio concepts to the ISM band. However, primary and secondary user concepts have been motivating and also convenient for the proposed scheme in this paper.

3 The Analysis of Aggregated ISM-PU Traffic

In this work, a testbed is designed to analyze the interaction between ISM-PU and ISM-SU nodes. It is well known that the IEEE 802.11 devices are the major interferers in the 2.4 GHz ISM band [9–11]. Therefore, as ISM-PUs, we use IEEE 802.11g wireless access points and laptops connected to these access points. As ISM-SUs, the 2.4 GHz IEEE 802.15.4-compliant hardware is used. Then, the RF Signal Strength (RFSS) measurements of the communication channel, which is the main factor for both Clear Channel Assessment (CCA) and Received Signal Strength Indicator (RSSI) values, were recorded by using the 2.4 GHz IEEE 802.15.4-compliant products in different RF channels.

In our test scenario, two IEEE 802.15.4-compliant products, one of which configured as a transmitter node and the other as a receiver node, communicate periodically at an interval of 10 ms. In this setup, the transmitter node measures two consecutive CCA values, and then transmits these measurements to the receiver node with a unique packet ID. The transmitter node logs the same packet on a computer via serial wired communication. If the packet is received seamlessly, the receiver node inserts calculated RSSI value to the packet and sends the packet to a computer via serial wired communication. Note that, packet loss rate can be obtained by comparing the packet ID logs of the transmitter and the receiver nodes. The measurements were done for two different scenarios: with an ISM-PU idle (no IEEE 802.11 interference) communication channel and with an ISM-PU busy communication channel, respectively. A schematic representation of the testbed is shown in Fig. 1.

The logged CCA and RSSI measurements are presented in Fig. 2. The results show that when the channel is ISM-PU idle, the mean of CCA measurements is −85 dBm and the variance is 5 dBm. Since the RSSI measurements are about −55 dBm, the calculated SIR value is approximately 30 dBm. For such SIR values, packet loss is

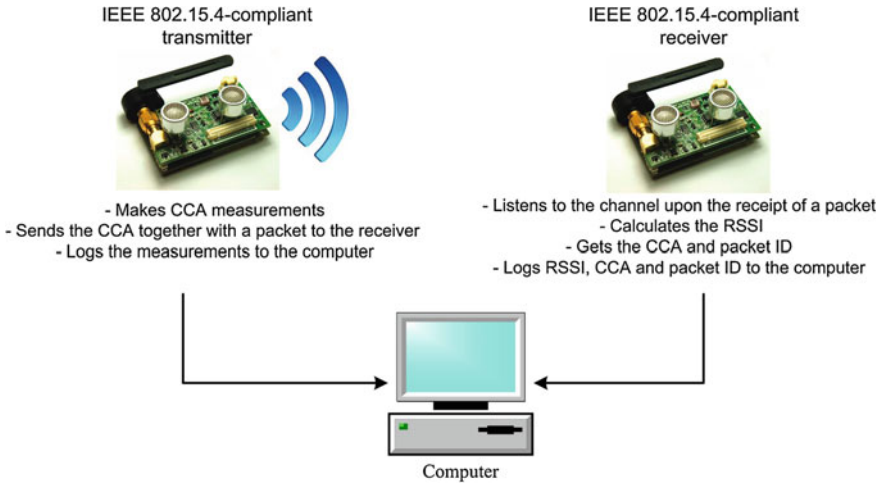


Fig. 1 A schematic representation of the testbed

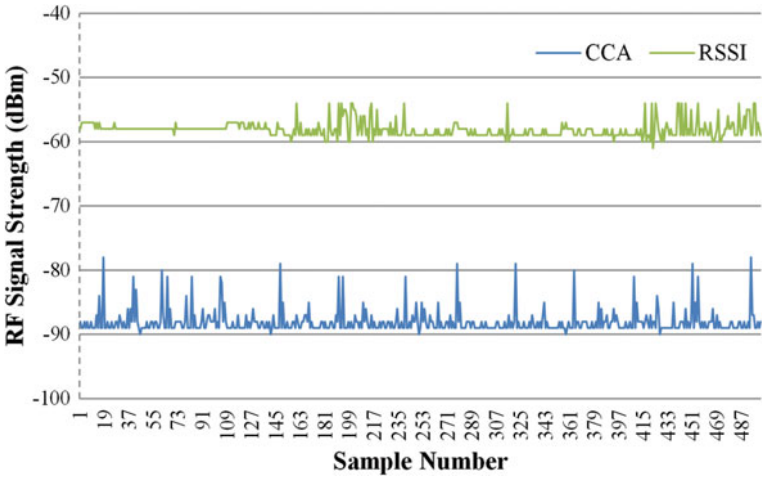


Fig. 2 500-sample snapshot of CCA and RSSI measurements for the ISM-PU idle scenario

not anticipated except for some particular environments. As expected, no packet loss is observed in the first scenario.

In the second scenario, called the ISM-PU busy scenario, the same testbed is used as in the first scenario. However, at this time, the ISM-PU devices around the laboratory are enabled. In this environment, neither the number of ISM-PUs nor the traffic pattern generated by them was manipulated. The traffic of the ISM-PU devices was a result of daily user activities in the offices around. Here, our goal is to examine the impact of real-world aggregated background IEEE 802.11 traffic on the IEEE

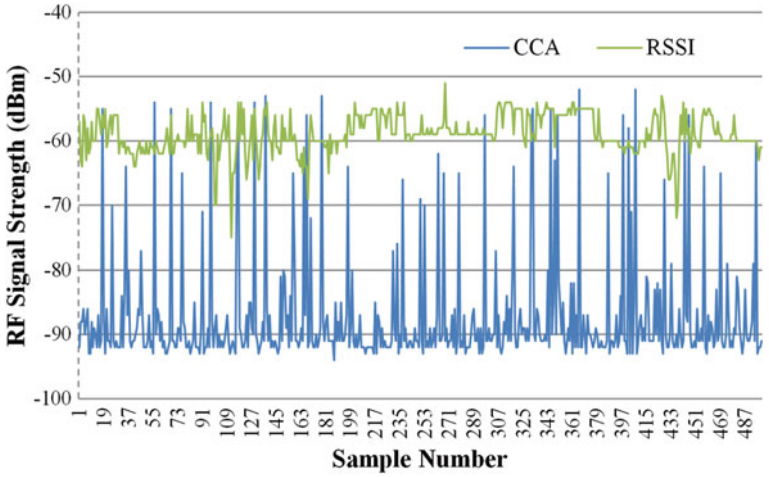


Fig. 3 500-sample snapshot of CCA and RSSI measurements for the ISM-PU busy scenario

802.15.4-compliant nodes. The CCA and RSSI measurements of the ISM-SU nodes are shown in Fig. 3. In these tests, the following points are observed:

- The CCA measurements, in the ISM-PU busy scenario, have high variance as seen in Fig. 3, which is due to the presence of ISM-PU traffic,
- Variations in the CCA values for different samples observed in the measurements is reasonable, since there are multiple ISM-PUs,
- A detailed analysis of the CCA and RSSI measurements reveal that only in 7.20 % of the observation period, there is ISM-PU traffic which forces SIR values to be below 10 dB and this causes a packet loss value of 6.94 %. Note that 10 dB threshold is selected because in the literature it is shown that 10 dB is a critical value for the IEEE 802.15.4 standard to have a successful communication [12].

Considering the fact that the communication channel is already ISM-PU idle for 92.8 % of the time, the high packet loss rate of 6.94 % indicates that the periodic channel access used in the testbed is not efficient enough.

4 Using High Autocorrelation Lag Values for Predicting the Aggregated ISM-PU Traffic Signals

In the previous chapter, the impact of co-existing Aggregated ISM-PU Traffic on ISM-SU is investigated using real-world RFSS traces. The testbed results showed that, in the environment, the ISM-SU packet loss ratio is close to the busy channel to free channel ratio. The main cause of such a result is that the channel access scheme cannot detect the co-existing interference literally. With this motivation, the

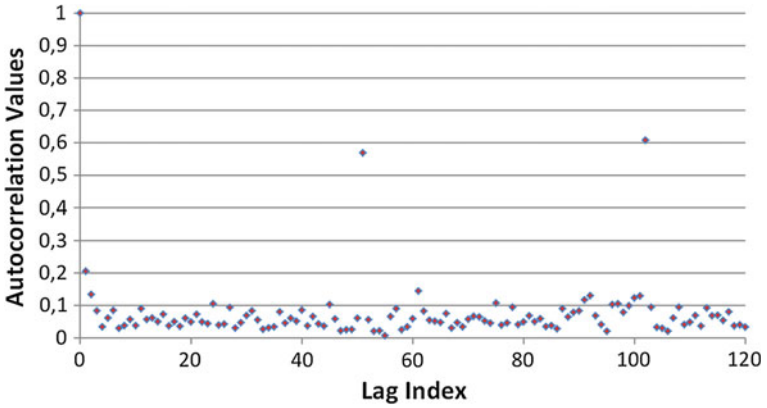


Fig. 4 Autocorrelation coefficients calculated for various lag values for ISM-PU busy scenario

CCA measurements gathered from the testbed are studied and then, a heuristic-based channel access scheme is proposed.

The periodic patterns of the co-existing traffic will be investigated in order to locate free channel access instants. In this approach, first, autocorrelation coefficients for varying lag values are calculated from CCA traces to detect the repeated patterns of the ISM-PU communication. The autocorrelation coefficients for the ISM-PU busy scenario are depicted in Fig. 4. This figure shows that the autocorrelation coefficients of some lag values are rather higher than the others. This is the cyclostationary feature effect of the ISM-PU communication. The proposed scheme assumes that for these high autocorrelation lag values, it is highly probable that there is co-existing ISM-PU traffic in the environment. At these instants, channel access of the ISM-SU should not be allowed.

It should be noted that there is more tha one ISM-PU in the environment. Thus, clustering these ISM-PUs according to the lag values would be beneficial for ISM-SUs to utilize the free bandwidth. However, as explained in the previous section, the ISM-SUs cannot measure the signal strength values of the ISM-PU communication properly which makes such a clustering impossible. Therefore, we propose to design a filter by using weighted autocorrelation lag values. This filter is going to be used by ISM-SU in order to predict the future ISM-PU interference.

In the proposed scheme the autocorrelation values, c_k, for different lags, k, are calculated as given in (1) whereas the maximum and the minimum signal strength values of the whole trace are given in (2) and (3), respectively.

$$C_k = \frac{\sum_{t=1}^{N-k}(q_t - \overline{q})(q_{t+k} - \overline{q})}{\sum_{t=1}^{N}(q_t - \overline{q})(q_t - \overline{q})} \qquad K = 1, \ldots, K \tag{1}$$

$$C_{\mathrm{MAX}} = |\max(c_k)| \qquad K = 1, \ldots, N \tag{2}$$

$$C_{\mathrm{MIN}} = |\min(c_k)| \qquad K = 1, \ldots, N \tag{3}$$

where q_t is the measured CCA value at index t, N is the total number of CCA samples and K representing the maximum lag value used in the algorithm. In order to determine the channel access instant for an ISM-SU, a simple filter, f_i, is designed as given in (4).

$$f_i = \frac{c_i - c_{\text{MIN}}}{c_{i\text{MAX}} - c_{\text{MIN}}} \qquad i = 1, \ldots, K \tag{4}$$

This filter value is applied to (5). For those values exceeding the control parameter T_{THR}, there is a probable ISM-PU traffic in the channel. At these instants where (5) returns busy channel, ISM-SU access to the channel should be prevented.

$$ChannelAcess(n + (i \times f_s^{-1})) = \begin{cases} free, & f_i \leq T_{\text{THR}} \\ busy, & f_i > T_{\text{THR}} \end{cases}, \quad i = 1, \ldots, N \tag{5}$$

where n indicates the current instant, f_s is the sampling frequency of the CCA measurements and T_{THR} is the decision threshold of the proposed scheme. For those values exceeding the control parameter T_{THR}, there is probable ISM_PU communication in the channel. Consequently, T_{THR} has a direct impact on the aggressiveness of the predictions made.

5 Performance Evaluation of the Scheme Proposed

The performance of the proposed scheme is compared with two other channel access schemes and results are given in Table. 1. In Scheme 1, every 10th sample on the test trace is selected. If the CCA value of this sample is below −75 dBm, this is recognized as a free channel access. Scheme 2 differs from Scheme 1 in terms of sample selection where the sample is selected according to uniform distribution within every 10-sample interval. Finally, in the proposed scheme, the channel access is scheduled to the instant with the lowest weight value.

It is obvious that the proposed scheme is highly capable of detecting ISM-PU traffic in the environment. However, we should keep in mind that there are instants where we predict ISM-PU traffic although the channel is free. This is the False Positive (FP) situation for our scheme. On the other hand, False Negative (FN) occurs

Table 1 Comparative free channel access performance of the proposed scheme

	Scheme 1	Scheme 2	Proposed scheme
# of free channel access	1851	1846	1979
# of busy channel access	149	154	21
% of free channel access	92.55 %	92.3 %	98.95 %
% of busy channel access	7.45 %	7.7 %	1.05 %

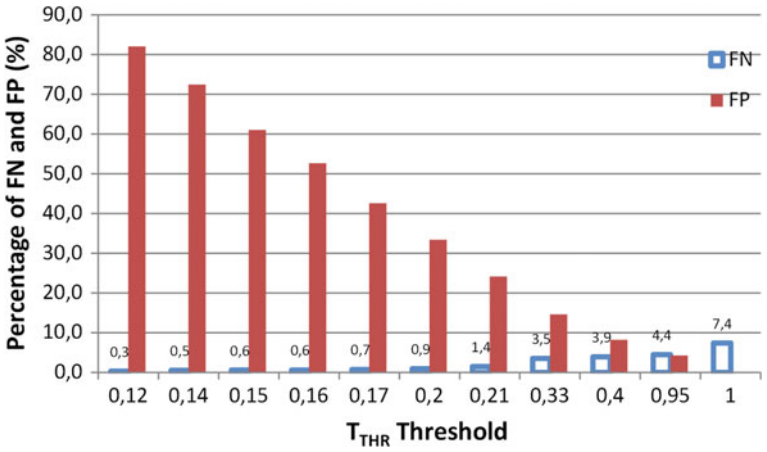


Fig. 5 The percentage of FNs and FPs for varying T_{THR} values

at the instants when the channel access is allowed although the channel is busy. In Fig. 5, the performance of the proposed scheme is evaluated based on the FN and FP rates of the channel access predictions for varying T_{THR} values where $N = 5000$, $K = 120$ and the ISM-PU traffic in the environment is occupying approximately 0.074 erlang (E). Figure 5 proves that free channel access prediction probability is inversely proportional to T_{THR}.

6 Conclusions

In this paper, 2.4 GHz ISM band aggregated traffic is analyzed, by using the adapted cognitive radio concepts. Then, a novel channel access scheme for the IEEE 802.15.4 devices is proposed. The performance of the introduced scheme is studied under various real-life IEEE 802.11 traffic logs. It is observed that the technique introduced gives promising results under the real world test traces, which motivates us to decrease the complexity of the scheme proposed and embed it to the IEEE 802.25.4 nodes as a future work.

References

1. J. Yun, B. Lee, J. Li, K. Han, A channel switching scheme for avoiding interference of between IEEE 802.15.4 and other networks, in *Proceedings 3rd International Multi-Symposiums on Computer and Computational Sciences (IMSCCS)*, pp. 136–139 (2008)
2. R. Musaloiu, A. Terzis, Minimising the effect of WiFi interference in 802.15.4 wireless sensor networks. Int. J. Sens. Netw. **2**(1), 43–54 (2008)

3. B.H. Jung, J.O.W. Chong, C.Y. Jung, S.U.M. Kim, D.K. Sung, Interference mediation for coexistence of WLAN and ZigBee networks, in *Proceedings 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications* (2008)
4. S. Myers, S. Megerian, S. Banerjee, M. Potkonjak, Experimental investigation of IEEE 802.15.4 transmission power control and interference minimization, in *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 294–303 (2007)
5. S. Pollin et al., Distributed Cognitive Coexistence of 802.15.4 With 802.11, in *Proceedings of IEEE CrownCom*, pp. 1–5 (2006)
6. L. Angrisani, M. Bertocco, D. Fortin, A. Sona, Experimental study of coexistence issues between IEEE 802.11b and IEEE 802.15.4 wireless networks. IEEE Trans. Instrum. Meas. **57**(8), 1514–1523 (2008)
7. W.Y. Lee, I.F. Akyildiz, Optimal spectrum sensing framework for cognitive radio networks. IEEE Trans. Wireless Commun. **7**, 3845–3857 (2008)
8. I.F. Akyildiz, W.Y. Lee, M.C. Vuran, S. Mohanty, Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey. Elsevier Comput. Netw. **50**(13), 2127–2159 (2006)
9. A. Sikora, V.F. Gora, Coexistence of IEEE 802.15.4 with Other Systems in the 2.4 GHz ISM Band, in *Proceedings of the IMTC Instrumentation and Measurement Tech. Conf*, vol. 3, pp. 1786–1891 (2005)
10. S.Y. Shin, H.S. Park, S. Choi, W.H. Kwon, Packet error rate analysis of zigbee under WLAN and bluetooth interferences. IEEE Trans. Wireless Commun. **6**(8), 2825–2830 (2007)
11. S. Pollin, I. Tan, B. Hodge, C. Chun, A. Bahai, Harmful coexistence between 802.15.4 and 802.11: A measurement-based study, in *Proceedings of CrownCom*, p. 1–6 (2008)
12. IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs): Coexistence analysis of IEEE Std 802.15.4 with other IEEE standards and proposed standards, white paper (2010)

A Parametric Study of CPN's Convergence Process

Antoine Desmet and Erol Gelenbe

1 Introduction

The Cognitive Packet network [1–4] is a QOS-oriented packet routing protocol, which is decentralised, distributed and adaptive. The CPN concept emerged in 1999, and has been applied to several network types: energy-constrained sensor networks [5], integration with IP networks [6], building graphs [7], etc., and to solve different problems access control [8], attack defense mechanisms [9], congestion management [10], emergency management [11] and more. A comprehensive overview of the work on CPN can be found in [12].

CPN has the particularity of being based on search techniques [13, 14] using dedicated exploratory packets, the "Smart Packets" (SPs), which explore various paths and let CPN maintain a current knowledge of the network. SPs can be issued by any node, and their next direction is chosen independently by each node they visit. A variety of algorithms exist to guide incoming SPs, and of particular interest for this paper, Random Neural Networks (RNNs) [15, 16]. RNNs "learn" which areas of the network are most worth exploring and direct SPs there. RNNs are hosted on each node and learn through reinforcement learning, with feedback provided by SPs which backtrack once they have identified a valid path.

This constant network exploration allows CPN to respond to changes in network conditions with low latency, yet without conducting extensive or systematic network searches, thanks to the intelligence provided by the RNNs. The performance and overhead-efficiency of CPN with RNNs is subject to several parameters. Of interest in this paper, is the influence of the "drift parameter", which controls the ratio of SPs forwarded according to the RNN, as opposed to randomly. To the best of our knowledge, all of the literature on CPN reports on trials of CPN for particular applications, but provide little to no information on how parameters are set [17, 18]. Our paper

A. Desmet (✉) · E. Gelenbe
Department of Electrical and Electronics Engineering, Intelligent Systems and Networks Group, Imperial College, London, UK
e-mail: a.desmet10@imperial.ac.uk

E. Gelenbe
e-mail: e.gelenbe@imperial.ac.uk

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_2

aims at addressing the gaps in the literature regarding the role played by each CPN parameter on the algorithm's performance.

In the following section, we present the CPN algorithm and RNNs in further detail. These concepts have been extensively covered in the literature, therefore we limit this introduction to the parts which are strictly relevant to our experiments. The third section analyses the effect of the drift parameter on CPN's performance during the initial knowledge buildup phase (convergence), using a bench-test setup. Finally, we draw conclusions on the overall influence of each metric and propose directions for future work.

2 CPN and RNN Operation

CPN relies on a small but constant flow of "Smart Packets" (SP) dedicated to network condition monitoring and new route discovery. SPs are routed on a hop-by-hop basis, where each node visited *independently* decides of their next direction. While a variety of algorithms can be used to guide SPs, the aim is to focus the stream of SPs in the most worthwhile areas of the network, while limiting random or complete graph searches. To prevent "lost" SPs from wasting resources, they are also given an upper limit on the number of nodes they can visit. Once a SP reaches its intended destination, it backtracks along its original path (with loops removed) and shares the information gathered along the way with every node. By gathering information from returning SPs, every node is able to build a "source-routing" route table, which is used to guide regular payload-carrying packets.

2.1 SP Routing

Smart Packets are routed on a hop-by-hop basis independently by each node. A variety of algorithms exist to determine the SP's next hop: they can be forwarded in randomly-chosen directions, and will perform a random walk of the network. The "Bang-Bang" [19] algorithm assumes an a priori knowledge of the network, which allows nodes to determine an "acceptable" range of route performance. If the known route falls within the "acceptable" range, SPs follow the best-known route; otherwise, the node promotes exploration by forwarding SPs at random. This approach is limited as it relies on a priori knowledge of the graph. Another approach is the Sensible routing policy [20] which forwards SPs probabilistically, based on the last path performance returned by each neighbouring node. The main challenge with this approach is to determine a mapping between path quality and corresponding SP forwarding probabilities. Approaches based on Genetic Algorithms have also been presented in the literature [21]. Another approach consists of running a Random Neural Network (RNN) [15, 16] in each node, and let it "learn" the areas most worth exploring and directs SPs there.

2.2 RNNs

Random Neural Networks [15, 16] are a form of neural networks inspired from biophysical neural networks, where, instead of having neuron activity defined as binary or continuous variable, it is defined as a *potential*, and nodes send "spikes" at random intervals to one another. RNNs have been applied to various problems, including task assignment [22], video and image compressing [23], and more. Each CPN node on the network runs an RNN, and a neuron is associated to each outgoing link. The node forwards incoming SPs to the node associated with the neuron with the highest level of excitation. Therefore, the global aim is to "train" the RNN so that the most excited neurons correspond to the neighbour nodes, which are most worth visiting. Training is done through reinforcement learning, using the information provided by successful SPs backtracking along their original path. An in-depth presentation of the RNN model and training process can be found in [15, 16]. CPN nodes occasionally disregard the RNN's advice and forward incoming SPs in a random direction to promote network exploration. This occurs when the RNN is in its initial state, before any feedback is provided by successful SPs. Past this initial phase, the CPN nodes also probabilistically forward SPs at random to prevent the RNNs from becoming overtrained (by sending the SPs on the same paths over and over). Forwarding a SP at random is referred as "drifting", and we define the *drift parameter* as the probability that an incoming SP will be forwarded according to the RNN's advice.

3 Experiments

In this section, we present a bench-test of the CPN algorithm, with an aim to reveal how the drift parameter influences CPN's performance. Our bench-test focusses on CPN's initial knowledge gathering phase: the "convergence" process. This experiment was conducted as part of a project where CPN was used as a routing algorithm for emergency building evacuations. The graph we use in this bench-test represents a three-storey building and consists of 240 nodes and 400 edges. Thus, CPN's aim is to identify a path from each node towards the nearest "sink": one of two exits located on the first floor. Our experimental protocol consists of letting all node send one SP (we refer this as a "batch" of SPs), and after each batch, we consult the routing table of each node: a score of $Q_P = 0$ is assigned if the node has not resolved a path yet. Otherwise, the score correspond to the ratio (expressed as percentage) of the shortest path's length (found using Dijkstra's algorithm) over the length of the path identified by CPN—so that the score is 100 % if CPN identifies the shortest path.

4 Results

Let us consider two trivial case-studies using the extreme values of the drift parameter to explore its expected influence on CPN's performance:

DriftParameter = 0. In this configuration, the nodes systematically disregard the RNN's advice and forward SPs at random. SPs effectively perform a random walk of the network, and thus any finite path has a non-zero probability of being visited through random walk. However, a path involving more nodes, or nodes with a higher degree is less likely to be visited.

DriftParameter = 1. Let us recall that a newly-initialised RNN which has not received any feedback has all its neurons in a "tie". This tie is broken by choosing one direction at random, thus SPs explore the network at random while searching for a valid path. As soon as a SP discovers a valid path, all neurons along that path receive positive reinforcement and become the most excited neurons. Because of the drift parameter setting, subsequent SPs will not drift and will follow the path of the most excited neurons (which corresponds to the path of the first successful SP), and further reinforce them indefinitely. This means CPN will only resolve one single path per departure node, and there are no guarantees on its optimality, since it was discovered through a random walk and is never further optimised through random exploration.

From these two case studies, we can infer that a low drift parameter guarantees that, given time, the optimal path will eventually be found, however the process might be extremely slow since the knowledge gathered by the RNN is disregarded. On the other hand, higher drift parameter values ensure a solution will be reached rapidly as information gathered by previous SPs is systematically re-used to guide the subsequent ones. However, this initial solution may be sub-optimal, and further improvement may be slow or limited because of an over-training phenomenon: the same sub-optimal path is reinforced over and over.

4.1 Overall Route Quality

Figure 1 illustrates the path resolution process for three representative values of drift parameter. The top graph illustrates the "slow-but-steady" learning process associated with low drift parameter values (i.e. mostly random SP movement): over a quarter of the departure nodes have no valid path by the second SP batch. At the end of the experiment, some nodes still have no path resolved, yet it is clear that CPN continues making small but continuous improvements at each step, as the box-plot gradually shrinks towards 100 %. On the other hand, the bottom graph associated to a high use of the RNNs exhibits the quickest initial path resolution process: CPN has resolved a path for most departure nodes after sending only one batch of SP. However, compared with the middle graph (drift parameter = 0.5), the median takes longer to reach 100 % in the long-term, and the box plots remain wider at the end

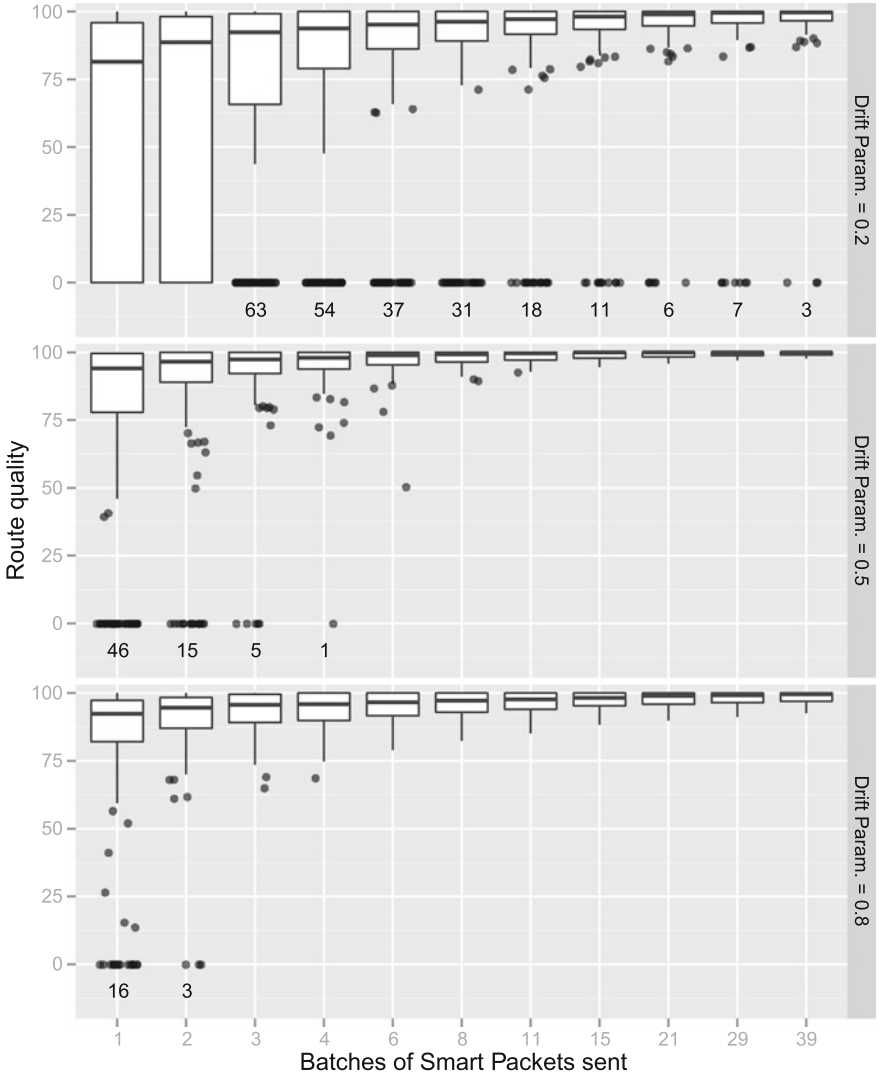


Fig. 1 Box-Plot showing the evolution in route quality based on the number of SPs sent. One sample for each 240 departure node. The *top* graph shows the "slow but steady" learning associated to low drift parameters, while the bottom one shows the "quick but approximate" discovery with high drift parameter

of the experiment. Past a fast learning phase, high drift parameter values lead to stagnation with sub-optimal values. Finally, the graph in the middle of Fig. 1 shows a "middle ground" where some of the initial resolution speed is "traded off" for a sustained higher improvement rate: while it takes five batches of SP for every node to acquire a path, the quality of the routes by the eighth SP batch is highest across

all experiments presented. This indicates that CPN's behaviour can be modulated through the drift parameter, to meet the needs of applications which place a higher emphasis on either convergence speed, or solution quality.

4.2 Spatial Convergence

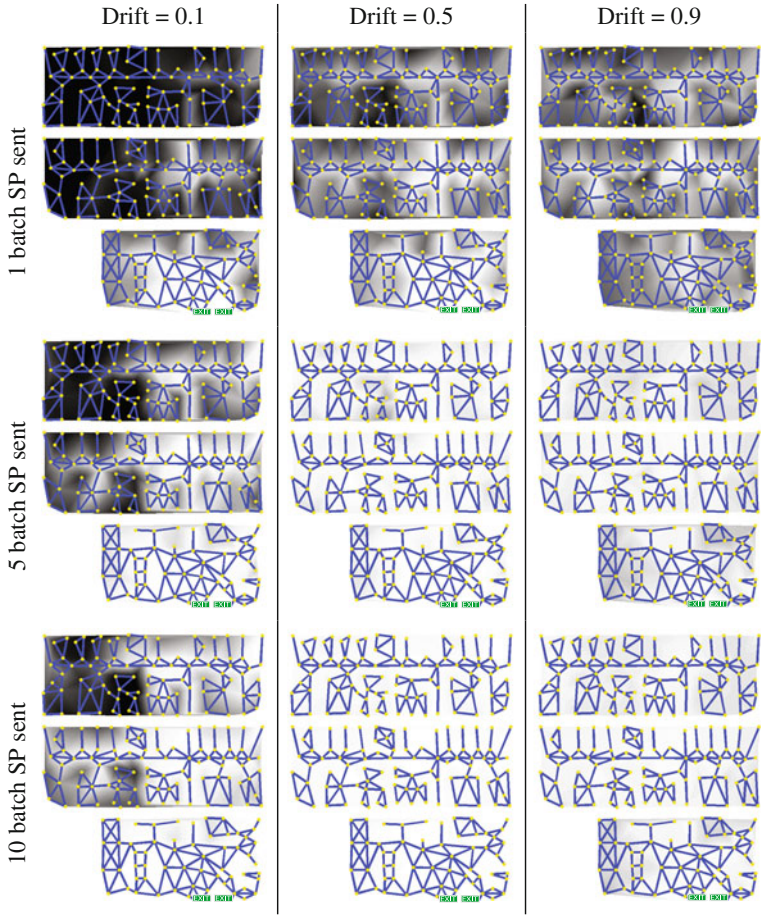
Having considered the overall convergence process of CPN, we now focus on its spatial features, to see if the convergence rate depends on the node's location. The figures arranged in Table 1 read as follows: vertically, each graph represents the path quality at three different stages of the convergence process: after {1, 5, 10} SP batches. Horizontally, we display the graphs for three drift parameters values: {0.1, 0.5, 0.9}. Each cell contains a "flattened" view of the building graph, with the ground floor at the bottom. The graph is colorised in shades of grey where black corresponds to $Q_P = 0\,\%$ (no path) and white to $Q_P = 100\,\%$ (optimal path). Beyond corroborating our previous findings, this graph shows that CPN node in the following areas converge at a faster rate:

Around the sinks The likelihood of identifying an path during a random search is increased for shorter paths, and therefore, SPs starting near the sinks are at an advantage. This explains why route quality globally resembles a gradient function, where the quality decreases with distance from the sinks.

Main routes We recall that the information gathered by a SP is not only available to the node which issued it, but is also shared with every node visited along the way. Therefore, while leaf nodes can only rely on "their own" SPs to gather information on the network, nodes located near the graph's backbone will harvest information from the many SPs transiting through them. This abundance of information helps them identifying the best path faster. The main corridors (horizontal edges across of the two upper floors) have a lighter shade than some other areas of the graph, which may be closer to the exits.

This experiment shows CPN rapidly establishes a "backbone" of high-quality paths, and then proceeds to explore more intricate areas. This ensures that packets will not have to travel too far before finding a node with a high-quality path to the sink. Our measurements show that CPN can be "tuned" to meet the needs of applications which can compromise on either solution quality, convergence speed or overhead. The analysis presented in this paper is still insufficient to make an informed decision on the optimal values of CPN's parameters. An extension to this research would consist of running *dynamic* bench-tests, where we would modify some edge's metric and measure the time taken by CPN to respond. We suspect that the drift parameter is likely to influence CPN's dynamic latency, as CPN will have to conduct a wider search of the graph to detect changes in metrics and identify alternative paths: this will be largely controlled by the drift parameter. Finally, there are some parameters which we have not considered, such as the *hop count limit* or the *damping coefficient*, further experimentation should be conducted to determine their effect on CPN's performance.

Table 1 Initial path resolution process across the three floors of the building



The lighter the colour the better the path found by CPN. The figure shows CPN starts by resolving a "backbone" of high-quality paths (main egress routes), and gradually progresses towards leaf nodes

References

1. E. Gelenbe, Cognitive packet network, U.S. Patent 6, 804,201 (2004)
2. E. Gelenbe, Z. Xu, E. Seref, Cognitive packet networks. In: 11th IEEE International Conference. Proceedings of Tools with Artificial Intelligence , pp 47–54 (1999)
3. E. Gelenbe, R. Lent, Z. Xu, Design and performance of cognitive packet networks. Perform. Eval. **46**(2), 155–176 (2001)
4. E. Gelenbe, R. Lent, Z. Xu, Towards Networks with Cognitive Packets, *Performance and QoS of Next Generation Networking* (Springer, London, 2001), pp. 3–17
5. L.A. Hey, Power aware smart routing in wireless sensor networks. In: Next Generation Internet Networks. NGI 2008, pp 195–202 (2008)
6. M.Gellman, Qos routing for real-time traffic. Ph D thesis, Electrical and Electronic Engineering Department, Imperial College London (2007)

7. H. Bi, A. Desmet, E. Gelenbe, *Routing Emergency Evacuees with Cognitive Packet Networks*, Lecture Notes in Electrical Engineering (Springer, Berlin, 2013)
8. E. Gelenbe, G. Sakellari, M. D'arienzo, Admission of qos aware users in a smart network. ACM Trans. Auton. Adapt. Syst. (TAAS) **3**(1), 4 (2008)
9. E. Gelenbe, M. Gellman, G. Loukas, An autonomic approach to denial of service defence. World of Wireless Mobile and Multimedia Networks, *WoWMoM 2005* in: Sixth IEEE International Symposium, pp. 537–541 (2005)
10. A. Desmet, E. Gelenbe, Reactive and proactive congestion management for emergency building evacuation. In: 38th Annual IEEE Conference on Local Computer Networks (LCN'13), Sydney, Australia, (2013)
11. A. Filippoupolitis, E. Gelenbe, A distributed decision support system for building evacuation. in: 2nd Conference on IEEE Human System Interactions. HSI'09. pp 323–330 (2009)
12. G. Sakellari, The cognitive packet network: a survey. The Computer Journal **53**(3), 268–279 (2010)
13. E. Gelenbe, Y. Cao, Autonomous search for mines. In: AeroSense'97, International Society for Optics and Photonics, pp 691–703 (1997)
14. E. Gelenbe, N. Schmajuk, J. Staddon, J. Reif, Autonomous search by robots and animals: A survey. Robotics Auton. Syst. **22**(1), 23–34 (1997)
15. E. Gelenbe, Random neural networks with negative and positive signals and product form solution. Neural Computation **1**(4), 502–510 (1989)
16. E. Gelenbe, Learning in the recurrent random neural network. Neural Computation **5**(1), 154–164 (1993)
17. E. Gelenbe, R. Lent, A. Nunez, Self-aware networks and QoS. Proc. of the IEEE **92**(9), 1478–1489 (2004)
18. G. Sakellari, E. Gelenbe, Adaptive resilience of the cognitive packet network in the presence of network worms. In: Proceedings of the NATO Symposium on C3I for Crisis, Emergency and Consequence Management, pp 11–12 (2009)
19. E. Gelenbe, E. Seref, Z. Xu, Simulation with learning agents. Proc. IEEE **89**(2), 148–157 (2001)
20. E. Gelenbe, Sensible decisions based on QoS. Comput. Manag. Sci. **1**(1), 1–14 (2003)
21. E. Gelenbe, P. Liu, J. Laine, Genetic algorithms for route discovery. IEEE Trans. Syst., Man, Cybern., Part B: Cybern. **36**(6), 1247–1254 (2006). doi:10.1109/TSMCB.2006.873213
22. Q. Han, Managing emergencies optimally using a random neural network-based algorithm. Future Internet **5**(4), 515–534 (2013)
23. C. Cramer, E. Gelenbe, P. Gelenbe, Image and video compression. IEEE Potentials **17**(1), 29–33 (1998)

Multi-cell Resource Block Allocation Framework

Fan Huang, Véronique Veque and Joanna Tomasik

Abstract We propose to combine the beamforming technique with Resource Block (RB) allocation algorithms to improve the performance in OFDMA networks. With MIMO antennas, the beamforming technique improves the received signal power which increases the RB's capacity and reduces the neighbouring cell users' interference. When the inter-cell interference channels are known, the beamforming parameters could be applied to the iterative scheduling methods to enhance the performance of the beamforming technique, hence increasing the total system throughput.

1 Introduction

In an LTE communication system, the Resource Block (RB) is the radio resource unit used by User Equipment (UE) to communicate with the Base Station, which is called evolved Node B (e-NodeB). Numerous RB allocation algorithms have been proposed to satisfy a specific parameter like the delay [1] or the throughput [3] while considering the single cell point of view. In these algorithms, the allocation decision is taken on the state of packets waiting in the UEs' buffers, the priority of a UE service and each UE channel capacity. In contrast to the first two parameters which are UE properties, the channel capacity depends mainly on the UE communication channels (received power) and inter-cell channels (inter-cell interference). In the Multiple Input Multiple Output (MIMO) system, the beamforming technique can be used to increase the received signal power and to decrease interference [6]. This technique varies the phase of the signals emitted by different antennas to enhance the performance of MIMO channels.

F. Huang (✉) · V. Veque
Laboratory of Signal and System, University of Paris-Sud, 91400 Orsay, France
e-mail: fan.huang@lss.supelec.fr

V. Veque
e-mail: veronique.veque@lss.supelec.fr

J. Tomasik
Department of Computer Science, SUPELEC, 91190 Gif-sur-Yvette, France
e-mail: joanna.tomasik@supelec.fr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_3

Interference control is a key challenge faced by designers of mobile communication systems. In traditional algorithms, the beamforming technique and other interference management techniques such as pre-coding, are not considered during the process of RB allocation, so the inter-cell interference does not change. This means that Signal to Interference and Noise Ratio (SINR) of each UE is not influenced by the RB allocation in its neighbouring cells [6].

In opposition to the traditional approach, the beamforming technique requires the introduction of interference management into RB allocation models. A beamforming parameter, called a beamformer, is applied to a frequency at the RB attributed to a UE. This parameter is responsible not only for the UE received power but also for interference in cells which are neighbours of the cell of the given UE [2]. This means that the SINR, which may be seen as equivalent to the capacity of the RB, changes on this frequency in the neighbouring cells, which consequently influences the RB allocation of these neighbouring cells. In this way, the RB allocation in all the cells interacts with the beamforming-OFDMA system.

In traditional methods, the RB allocation and the beamforming procedure are performed separately in different entities:

1. RBs are allocated to UEs.
2. When UEs get an RB, the system executes a beamforming procedure to increase the RB capacity.

We consider that these two steps are dependent: the RB allocation is determined by the UE's capacity, but the capacities are also influenced by the beamforming technique. The decision of RB allocation and beamforming should be determined jointly to improve the performance of multi-cell networks.

We propose an iterative method to allocate an RB and to find the best beamforming parameter at the same time.[1]

The rest of this paper is organized as follows. Section 2 introduces the single cell version RB allocation algorithm and the beamforming techniques, defines the multi-cell network model with the beamforming technique. Section 3 deals with the problem of the multi-cell RB allocation, using concepts such as an iterative method and two-level optimization. Section 4 presents the simulation results. Section 5 provides conclusions and ideas for further research in this area.

2 Work Related to the System Model

The main RB allocation algorithms [1–4] consider the single cell point of view and aim at optimizing a specific parameter like delay or throughput depending on channel capacity. The channel quality is highly time-varying and is given by the channel state

[1] SYSTEMATIC PARIS-REGION and the SOAPS.2 project are supported by the French Ministry of Industry, the Île-de-France Regional Council, the department of Essonne, and the department of Yvelines.

information (CSI) in an LTE system [5]. The Maximum Throughput (MT) [1] is the basic algorithm. It computes the UE capacity on each RB and allocates the RB, whose capacity is maximal, to the UE. We use this algorithm as an example to present our framework, and start by describing the classical approach to interference control with the use of beamforming.

The operator allocates the RBs in a multi-cell network made up of a central cell and $J-1$ neighbouring cells with the same assigned frequency. There are I UEs in each cell. Let j denote any of these J cells, and i_j denote a UE in the jth cell. The working frequency contains M RBs. As all the sub-carriers are orthogonal, the inner-cell interference is supposed to be equal to zero. SINR is influenced by the attribution of RBs on the same frequency in the network and their corresponding beamforming vectors.

To change the directionality of the array when transmitting, a beamformer controls the phase and the relative amplitude of the signal at each transmitter. Assuming the phase and the amplitude of the first antenna have the standard value, and the phase from the other antennas has $[-\pi, \pi]$ difference with the first antenna, the amplitude is $(0, \sqrt{N-1})$ times that of the first antenna, where N is the number of antennas. The phase and the amplitude of signals from multi-antennas could be presented by beamforming vectors: $\frac{1}{\sqrt{N}}[|r_1|e^{i\phi_1}, ..., |r_n|e^{i\phi_n}, ...]$, where $|r_n|$ represents the amplitude of the signal from the nth antenna and ϕ_n stands for the phase difference.

Our improvement of the RB allocation algorithm is based upon the cooperation with the beamforming. In the traditional multi-cell network, the choice of beamforming vectors is based on two factors: the received power and the neighbouring cells' interference. Some beamforming schemes are introduced [7–10] to find a trade-off between increasing the received signal power and reducing interference.

The eNode-B in each cell is located at the centre and contains N_{T} transmitting antennas. It is assumed that each UE has N_{R} receiving antennas. During the reception of a signal from the e-NodeB, a UE suffers from inter-cell interference from its neighbouring cells on the same frequency. The SINR combining at the UE i_j in the jth cell is expressed as in [10]:

$$\mathrm{SINR}_{i_j}^{m} = \frac{P||H_{j,i_j}\omega_j^m||^2}{\sum_{k\in\Phi(j)}|\omega_j^{m*}H_{j,i_j}^{*}H_{k,i_j}\omega_k^m| + \sigma_0}, \tag{1}$$

where $H_{k,i_j} \in \mathbb{C}^{N_{\mathrm{R}}\times N_{\mathrm{T}}}$ denotes the communication channel matrix between the e-NodeB in the kth cell and the UE i_j, $\omega_j^m \in \mathbb{C}^{N_{\mathrm{T}}}$ denotes a transmit beamforming vector of the e-NodeB in the jth cell on the frequency of RB m, $\Phi(j)$ indicates a group of neighbouring cells around the jth cell, and σ_{i_j} represents a white Gaussian noise vector at UE i_j with the power of each entry σ_0. The average signal power is noted as P. We assume that each UE perfectly knows the CSI value of its communication channel H_{j,i_j} and interference channel H_{k,i_j}, $k \in \Phi(j)$ based on the feedback channel. Each e-NodeB selects its transmit beamforming vector ω_j^m according to the

same codebook $\mathbb{F}$ which is defined in [10]. Hence, SINR at the UE side in the jth cell on the frequency of RB m depends on the transmit beamforming vectors in each cell.

The modifications which we propose to introduce into this model are explained in detail in the next section.

3 Proposed Multi-cell RB Allocation with Beamforming

In the conventional RB allocation methods, the beamforming vectors are chosen after the RB allocation. In our optimization problem, the RB choice is also a function of beamforming vectors $i_j(\omega_1^m, ..., \omega_J^m)^*$. When beamforming vectors change the allocation is different because the UE's capacity varies as a result of SINR, according to Eq. (1).

On the other hand, the beamforming vectors are also determined by the UEs' channel quality, because the vectors $\{\omega_1^*, \omega_2{}^*, ..., \omega_J^*\}(i_1, ..., i_J)$ are also a function of UE's channel matrix H.

The classical solution based on the greedy approach is to allocate the RB having the highest capacity to a UE [1]. Put differently, we find the UE with the highest SINR in each cell, and we make an allocation decision for them on RB m: $\mathbb{P}_0^m = (i_1, ..., i_J)$ a group of UEs which gets this RB in each cell. After the RB allocation, we apply the beamforming to the antennas to increase the channel capacity for each chosen UE. Denoting the set of beamforming vectors as $\mathbb{W}_0^m$, which contains $(\omega_1^m(0), ..., \omega_J^m(0))$. With the allocation decision and beamforming vectors, we can estimate the total system throughput R_0^m on RB m.

Because the scheduler in each cell will influence the neighbouring cell interference, the sum of each cell's maximum throughput is not necessarily equivalent to the maximum of the total network throughput.

We should enumerate over all combinations of the UE channels and beamforming vectors to find the expected maximum total network throughput. The computation complexity is thus exponential, $\mathbb{O}(I^{\mathbb{F}})$, where $||\mathbb{F}||$ is the number of elements in the beamforming codebook $\mathbb{F}$. For this reason, the exact allocation solution cannot be obtained in a reasonable time.

We propose a heuristic method of complexity in $\mathbb{O}(I \times ||\mathbb{F}||)$. The orders of the iterative schemes which we propose to compute interference are depicted in Fig. 1. These two schemes will be applied to the next two algorithms, and the difference between them is analyzed and compared in Sect. 4.

We define two loops in our modified algorithm. In the internal loop, we allocate the RBs in J cells, starting from cell 1 until J; in the external loop, we repeat the internal loop to improve the allocation. Our improvement of the RB allocation algorithms is based upon the cooperation of the beamforming technique.

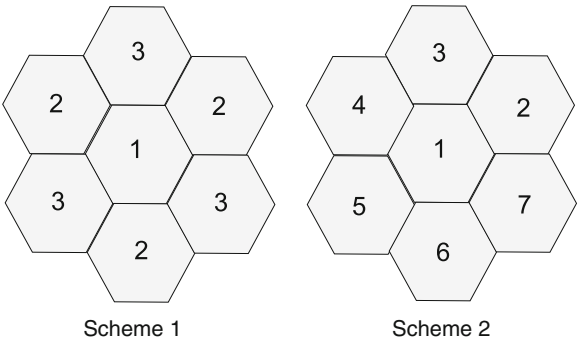


Fig. 1 Multi cell enumeration schemes

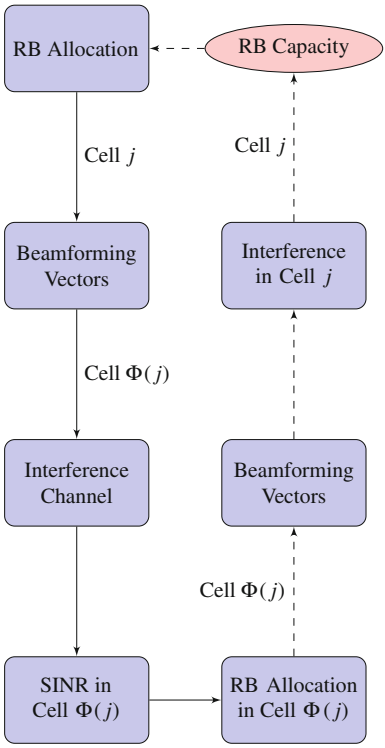


Fig. 2 Traditional process diagram

3.1 Internal Loop

The allocation of RBs is made sequentially cell after cell. In the first cell, our only objective is to maximize the SINR on RB m:

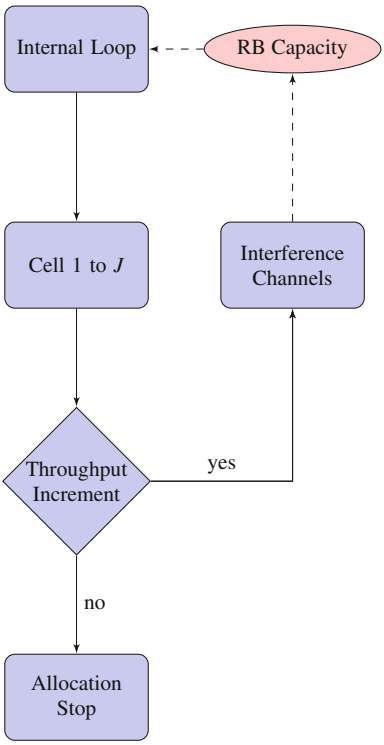


Fig. 3 Modified algorithm diagram

$$\max_{i_j,\omega_j^m} \frac{P||H_{j,i_j}\omega_j^m||^2}{I_{i_j}(\mathbb{W}_{j-1}^m(l)) + \sigma_0}, \tag{2}$$

where

$$I_{(i_j)}(\mathbb{W}_{j-1}(l)) = \frac{P}{||H_{j,i_j}\omega_j^m(l)||} \sum_{\omega_k^m(l)\in\mathbb{W}_{j-1}(l)} |\omega_j^{m*}(l)H_{j,i_j}^* H_{k,i_j}\omega_k^m(l)|, \tag{3}$$

UE i_1^* which has the highest SINR gets the RB. The corresponding beamforming vector is $\omega_1^{m*}(l)$. We denote the new allocation decision on RB m as $\mathbb{P}_1^m(l) = (i_1^*, i_2, ..., i_J)$ and beamforming set as $\mathbb{W}_1^m(l) = (\omega_1^{m*}(l), \omega_2^m(l-1), ..., \omega_J^m(l-1))$, where l is the number of the external loop executions as described in Sect. 3.2. At the lth loop, we estimate the total system throughput $R_1^m(l)$, compare it with traditional system throughput R_0^m. If step $R_1^m(l) \geq R_0^m$, we apply the new allocation policy and beamforming vectors in the 1st cell. If $R_1^m(l) < R_0^m$, we keep the traditional solution (as shown in Fig.. 2): $\mathbb{P}_1^m(l) = \mathbb{P}_0^m$ and $\mathbb{W}_1^m(l) = \mathbb{W}_0^m$.

For the jth cell ($j = 2, 3, ..., J$), we consider the beamforming vectors of the former $j-1$ cells. Thanks to Eq. (2) we can obtain the allocation policy $\mathbb{P}_j^m(l)$ and

beamforming set $\mathbb{W}_j^m(l)$. As before, we compare the estimated data rate $R_j^m(l)$ and $R_{j-1}^m(l)$, if $R_j^m(l) \geq R_{j-1}^m(l)$, the final allocation policy is $\mathbb{P}_j^m(l)$ and beamforming set is $\mathbb{W}_j^m(l)$. Otherwise we continue to use the former allocation policy $\mathbb{P}_{j-1}^m(l)$ and beamforming vector $\mathbb{W}_{j-1}^m(l)$. An internal loop run will thus produce an allocation at least as good as a classical one. The numerical complexity of this algorithm is $\mathbb{O}(I \times ||\mathbb{F}||)$.

3.2 External Loop

When the internal loop terminates at cell J, we get a new RB allocation policy $\mathbb{P}_J^m$ and beamforming vector $\mathbb{W}_J^m$. Based on the new beamforming vectors in the neighbouring cells, we may allocate this RB to another UE and the corresponding beamforming vector to increase the system throughput. So we repeat the internal loop (as shown in the Fig 3):

Modified Multi-Cell Maximum Throughput Algorithm
Initialize $l \leftarrow 1$, set $\mathbb{P}_0^m = \mathbb{P}_J^m(l)$, $\mathbb{W}_0^m = \mathbb{W}_J^m(l)$
repeat
$l \leftarrow l + 1$
Repeat the internal loop from cell 1 to J
Estimate system throughput $R_J^m(l)$ on RB m
Update the new $\mathbb{P}_0^m$ and $\mathbb{W}_0^m$
until $R_J^m(l) \leq R_J^m(l-1)$

After the external loop, we get the final allocation policy $\mathbb{P}_J^m(l)$ and beamforming vector $\mathbb{W}_J^m(l)$. Here we give a proof for the convergence of our algorithm.

Proof From the definition of the algorithm, we can conclude that the system throughput is not decreasing in each loop:

$$R_j^m(l) \geq R_{j-1}^m(l),\ R_j^m(l) \geq R_j^m(l-1), \tag{4}$$

The RB capacity function $R(\text{SINR})$ is a step function [2] so the system throughput function $R_j^m(l)$ is also a step function. The property of the step function is: if $R_j^m(l) > R_j^m(l-1)$, $R_j^m(l) - R_j^m(l-1) > \Delta$, where Δ is the minimum difference between two neighbouring steps of $R(\text{SINR})$.

Because the white noise is always presented, we can estimate an upper bound R_{upper}^m for the system throughput on this frequency by ignoring the interference:

$$R_{\text{upper}}^m = \max_{\mathbb{P},\mathbb{W}} \sum_{j=1}^{J} \frac{P||H_{j,i_j}\omega_{j(j)}^m||^2}{\sigma_0}. \tag{5}$$

The number of loops L satisfies that:

$$L < \frac{R^m_{\text{upper}} - R^m_0}{\Delta} \tag{6}$$

From Eqs. (4) and (6), we can conclude that our multi-cell maximum throughput algorithm is convergent.

Because the convergence of the algorithm, the algorithm stops after several times of internal loop, the complexity remains at the level of $\mathbb{O}(I \times ||\mathbb{F}||)$.

4 Numerical Results

In this section, we compare the performance of our modified multi-cell RB allocation algorithm with its single cell version. Our performance evaluation is made using LTE-Sim [11], a simulation package which provides a library of all LTE functions and architecture. We modified the source code of this simulator and added the MIMO channel model to complete our simulation.

In our simulation setup, the bandwidth of the LTE network is 5 MHz and 25 RBs are available at the same time [5]. The radius of the cell is 1 km and the number of UEs per cell increases gradually from 20 to 60. The UEs are uniformly distributed in the cell by the distribution function of LTE-Sim [11]. They are divided into two groups, the centre group includes 50 % of the UEs who are nearest to the e-NodeB, the edge group is the rest of the UEs. The Transmission Time Interval (TTI) is fixed to 1 ms, and an individual simulation time run lasts 100 s. We perform a series of simulation runs in order to obtain results on a confidence level $\alpha = 95\,\%$. We repeat simulations 100 times, which is sufficient to get the confidence level assumed and calculate the average value and variance of each performance criterion. Because the channel is fast-fading and simulation time is long, the confidence intervals are very small and for this reason we do not mark them on the Figs. 4 and 5.

We run simulations for the MT algorithm with the infinite buffer service because it is used to maximize the system throughput.

We first compare the throughput improvement of our modified algorithms. Our method improves the effect of beamforming, so Figs. 4 and 5 show that the RB capacity is higher than the conventional algorithm, the average throughput per UE is about 10 % higher on the cell edge and 5 % in the cell centre.

The interference is higher on the cell edge than in the centre, and our proposed method decreases the interference significantly, so the average throughput per UE on the cell edge group increases more than in the cell centre. In the centre of the cell, the interference is exponentially reduced because of the path-loss, so the beamforming technique is mainly used to increase the received signal power, hence the enhancement is not as high as on the cell edge.

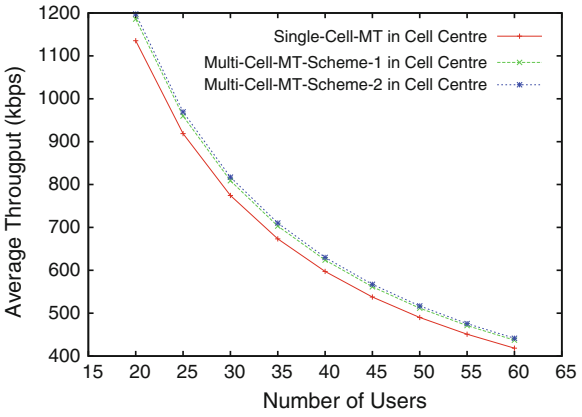


Fig. 4 Cell centre throughput

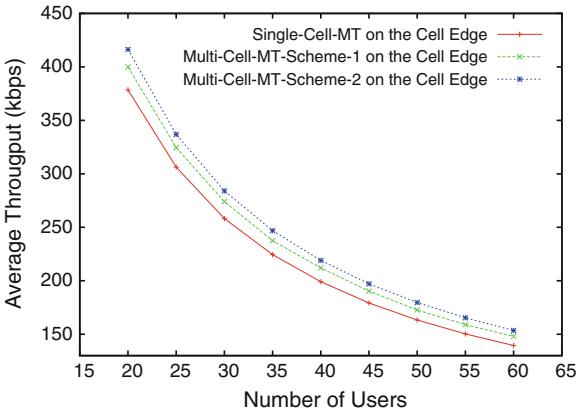


Fig. 5 Cell edge throughput

Our modification increases the throughput in two ways:

1. Choosing the best $||H_{j,i_{j,h}}\omega_j^m||$ to increase the received signal power, enumeration schemes in Fig. 1 benefit from this advantage, and for this reason, they have the same improvement in the cell centre.
2. Adjusting the inter-cell channels $\sum_{k\in\Phi(j)}||\omega_j^{m*}H_{j,i_{j,h}}^*H_{k,i_{j,h}}\omega_k^m||$ to limit the interference, in scheme 1 we consider the interference channel information from fewer cells than in scheme 2. For this reason the interference management effect in scheme 1 is worse compared to scheme 2. This effect leads to a less significantly improvement in the large interference area, namely on the cell edge.

Figure 6 presents the average improvement of each external loop in our modified algorithm. The throughput improves greatly in the first passage of the external loop.

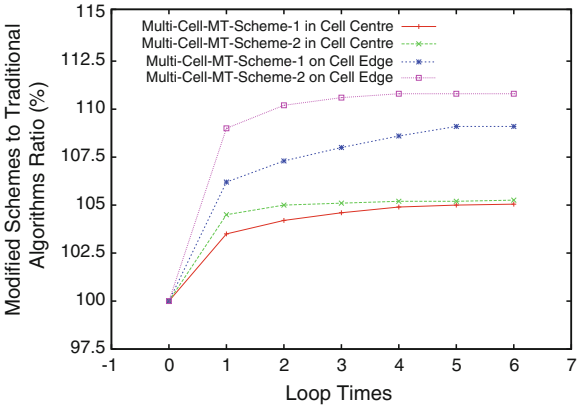


Fig. 6 Allocation schemes alteration

The further loop executions do not increase the throughput significantly. Simulation makes us observe that the algorithm converges quickly and throughput stabilizes after five or six repetitions.

5 Conclusions and Perspectives

In this paper, we presented algorithm of RB allocation working with the beamforming technique for multi-cell networks. We took into account the interference resulting from the RB allocation performed in the neighbouring cells. This increases the channel's efficiency by adjusting the interference of each cell. We proposed an iterative method to reduce the computation complexity and to improve network performance. Compared to single cell version algorithms, more control channel information should be shared among all the cells.

Our method has an enhancement of about 5 % in the cell centre and 10 % on the cell edge. It can also be applied on the other RB allocation algorithms like EXP Rule and Log Rule. With a polynomial complexity, the network considerably benefits from our mechanism on the cell edge, which is obviously the most prone to suffer from interference. This feature is particularly important in security wireless infrastructures because they must meet very high QoS (Quality of Service) requirements.

We analyzed different cell orders according to which beamforming control messages are transmitted. We kept the reduction of this overhead in mind. Comparing enumeration schemes 1 and 2, more sharing of channel information leads to better performance. Scheme 2 has better improvement on the cell edge because it has more neighbouring cell information to reduce interference. To extend the J-cell network to one with an infinite number of cells, scheme 1 is more adapted because it needs less neighbouring cell beamforming information.

Our future work is to incorporate cooperation among cells during the RB allocation to enhance performance without increasing complexity.

References

1. B.S. Tsybakov, File transmission over wireless fast fading downlink. IEEE Trans. Inf. Theory **48**, 2323–2337 (2002)
2. H.A.M Ramli, R. Basukala, K. Sandrasegaran, R. Patachaianand, Performance of Well Known Packet Scheduling Algorithms in the Downlink 3GPP LTE System, in *2009 IEEE 9th Malaysia International Conference on Communications (MICC)*, pp. 815–820 (2009)
3. S. Shakkottai, A. Stolyar, Scheduling for Multiple Flows Sharing a Time-varying Channel: The Exponential Rule, Amer. Mathematical Soc. Translations, Series 2 (a volume in memory of F. Karpelevich), Y. M. Suhov, Ed., vol. 207 (2002)
4. G. de Veciana, B. Sadiq, S. JunBaek, Delay-optimal opportunistic scheduling and approximations: the log rule. IEEE/ACM Trans. Netw. **19**(2), 405–418 (2011)
5. S. Sesia, M. Baker, I. Toufik, LTE - The UMTS Long Term Evolution: From Theory to Practice Wiley Press, London (2009)
6. R.S. Blum, MIMO capacity with interference. IEEE J. Sel. Areas Commun. **21**(5), 793–801 (2003)
7. D.J. Love, R.W. Heath, T. Strohmer, Grassmannian beamforming for multiple-input multiple-output wireless systems channels. IEEE Trans. Info. Theory **49**(10), 2735–2747 (2003)
8. T. Yoo, A. Goldsmith, On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming. IEEE J. Sel. Areas Comm. **24**(3), 528–541 (2006)
9. J. Zhang, J.G. Andrews, Adaptive spatial intercell interference cancellation in multicell wireless networks. IEEE J. Sel. Areas Comm. **28**(9), 1455–1468 (2010)
10. I. Sohn, S.H. Lee, J.G. Andrews, Belief propagation for distributed downlink beamforming incooperative MIMO cellular networks, IEEE Trans. on Wireless Comm. vol. 10, no. 12, 4140–4149 (2011)
11. G. Piro, L. Grieco, G. Boggia, F. Capozzi, P. Camarda, Simulating LTE cellular systems: an open source framework. IEEE Trans. Veh. Tech. **60**(2), 498–513 (2010)

An Implementation of Voice Over IP in the Cognitive Packet Network

Lan Wang and Erol Gelenbe

Abstract Voice over IP (VOIP) has strict Quality of Service (QoS) constraints and requires real-time packet delivery, which poses a major challenge to IP networks. The Cognitive Packet Network (CPN) has been designed as a QoS-driven protocol that addresses user-oriented QoS demands by adaptively routing packets based on online sensing and measurement. This paper presents our design, implementation, and evaluation of a 'Voice over CPN' system where an extension of the CPN routing algorithm has been developed to support the needs of voice packet delivery in the presence of other background traffic flows with the same or different QoS requirements.

Keywords Cognitive packet network · Voice over CPN · Quality of service · Experimental implementation

1 Introduction

In the current 'All IP' era, IP networks are required to guarantee Quality of Service (QoS) for a vast variety of communication services and users [1], especially for real-time voice services, which have stringent QoS constraints. In the past decade, many QoS approaches such as IntServ&RSVP, DiffServ and MPLS were proposed, yet with limited effect. While networks must be monitored and measured [2, 3] for performance in terms of loss [4], delay, packet desequencing [5], for topology purposes [6] and reliability [7], mathematical models are also used [8] together with measurements and simulations.

L. Wang (✉) · E. Gelenbe
Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College London, London, UK
e-mail: lan.wang12@imperial.ac.uk

E. Gelenbe
e-mail: e.gelenbe@imperial.ac.uk

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_4

The Cognitive Packet Network (CPN) has been designed as a QoS-driven protocol that addresses user-defined QoS demands by routing packets in the manner that adapts to the varying network conditions based on online sensing and measurement [9–11]. It was developed under the open-source platform Linux, incorporated into IP protocols and running as a loadable kernel module at any linux machine, which made it feasible to construct a large and measurable CPN testbed carrying many network services. It has also been used for organised exit paths for emergency management operations [12, 13].

In CPN, QoS requirements specified by users, such as *Delay*, *Loss*, *Energy* [14, 15], or a combination of the above, are incorporated in the 'goal' function, which is used for the CPN routing algorithm. Three types of packets are used by CPN: smart packets (SPs), dumb packets (DPs) and acknowledgments (ACKs). SPs explore the route for DPs and collect measurements; DPs carry payload and also conduct measurements; ACKs bring back source routing information for the DPs. SPs discover the route using random neural network (RNN)-based reinforcement learning (RL) [16–18], which resides in each node. Each RNN in a node corresponds to a QoS class and destination pair [19] with each neuron representing a outgoing link from the node. The arrival of an SP for a specific QoS class at a node triggers the execution of the RNN algorithm with the weights updated by Reinforcement Learning (RL) using QoS goal-based measurements, whereby routing decision are based on selecting the output link corresponding to the most excited neuron [20, 21].

At the sender in a VOIP application installed at a CPN node, the original analog voice signals are sampled, encoded and then packetised into IP packets by adding RTP header, UDP header and IP header. These packets travel across the IP network employing the CPN protocol, where IP-CPN conversion is performed at the source node by encapsulating IP packets into CPN packets. Previous research has reported the ability of CPN to alleviate delay, jitter, packet desequencing and packet loss for real-time traffic by smartly selecting the path that provides the best possible QoS required by a user (or an application) [20, 23], and thus the packets in a voice traffic flow may traverse different paths over the CPN network. At the receiver, packets are queued in a buffer to reduce jitter while reordering algorithm and loss concealment techniques are applied before the recovery of the original voice signals. Packets that arrive later than the required playback time or those that provoke buffer overflow, are discarded, contributing to the end-to-end packet loss.

This paper presents an extension of CPN structured as in Fig. 1, to support the QoS of voice packet delivery in the presence of other background traffic flows with the same or different QoS requirements. The resulting system was evaluated in order to select the QoS goal that provides better performance for all network load conditions. Furthermore, we study the correlation of voice packet end-to-end loss with path switching induced by the adaptive scheme that is inherent to CPN.

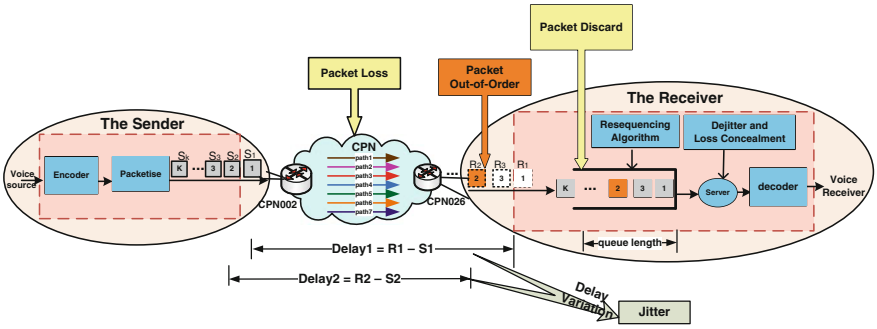


Fig. 1 VOCPN system structure [22]

2 Voice Over CPN Supporting Multiple QoS Classes

This section presents the incorporation of 'Jitter' minimisation into the goal function used by CPN in order to match the needs of voice delivery, as well as the extension of CPN that supports multiple QoS classes for multiple traffic flows simultaneously. In RFC3393 and RFC5481 , 'Packet Delay Variation' is used to refer to 'Jitter'. One of the specific formulations of delay variation implemented in the industry is called Instantaneous packet delay variation (IPDV), which refers to the difference in packet delay between successive packets, where the reference is the previous packet in the stream's sending sequence so that the reference changes for each packet in the stream.

The measurement of IPDV for packets consecutively numbered $i = 1, 2, 3, \ldots$ is as follows. If S_i denotes the departure time of the ith packet from the source node, and R_i denotes the arrival time of the ith packet at the destination node, then the one-way delay of the ith packet $D_i = R_i - S_i$, and IPDV is

$$IPDV_i = |D_i - D_{i-1}| = |(R_i - S_i) - (R_{i-1} - S_{i-1})| \tag{1}$$

To fulfill the QoS goal of minimising jitter, online measurement collects the jitter experienced by each dumb packet. Since in CPN each DP carries the time stamp of its arrival instant at each node along its path, so when a DP say DP_i arrives at the destination, an ACK is generated with the arrival time-stamp provided by the DP. As ACK_i heads back along the inverse path of the DP, at each node the forward delay $Delay_i$ is estimated from this node to the destination by taking the difference between the current arrival time of ACK_i at the node and the time at which the DP_i reached the same node [20], divided by two. This quantity is deposited in the mailbox at the node. The instantaneous packet delay variation is computed as the difference between the value of $Delay_i$ and $Delay_{i-1}$ of the previous packet in the same traffic flow as defined in (1), and jitter is approximated by the smoothed exponential average of IPDV with factor a smoothing factor 0.5:

$$\overline{J_i} = \frac{J_{i-1}}{2} + \frac{J_i}{2} \tag{2}$$

Then, the updated value of jitter is deposited in the node's mailbox. When a subsequent SP for the QoS class of Jitter and the same destination enters the node, it uses the value from the mailbox to compute the reward $Reward_i$ and in turn update the weights of the corresponding RNN, which is then executed to decide the outgoing link [20].

$$\text{Reward}_i = \frac{1}{\overline{J_i} + \epsilon} \tag{3}$$

where ε is used to ensure the denominator is non-zero.

We enable CPN to support multiple QoS classes simultaneously, for multiple flows that originate at any node and each flow is routed based on its specific QoS criteria, the following steps are needed: (1) The traffic differentiation is conducted relying on source MAC (or IP), destination MAC (or IP) and the TCP/UDP port of the applications. For instance, the VOIP application 'Linphone' originates voice packets with its dedicated SIP port (5060) and audio port (7080) residing in the fields of the UDP header. (2) The QoS class definition is based on the QoS requirements of different users or applications, which is configurable and loaded into the memory, while CPN is being initiated. (3) CPN treats each traffic flow according to its QoS class using multiple RNNs at each node, where each RNN corresponds to a QoS class and a source-destination pair.

3 Path Switching, Packet Reordering and Loss

CPN adaptively selects the path that provides best possible QoS requested for traffic transmission, leading to possible path switches. Traffic may suffer packet desequencing and loss if paths switch excessively. Accordingly, we are interested in examining the correlation between undesirable effects such as packet desequencing and end-to-end loss, and path switching. In the following sections, we described methods to carry out measurements and statistics for the three metrics.

For a given flow in CPN, the routing information explored by SPs is encapsulated into the CPN header for each DP as it originates from the source node, whereby the path used by each DP can be detected. The metric we are interested in is the 'Path Switching Ratio', which is defined as the number of path switches (Q_{path}) in a given flow during the time interval being considered being divided by the total number of packets forwarded (N), as well as the 'Path Switching Rate' ($\text{Rate}_{\text{path}}$) defined as Q_{path} being divided by the time interval (T).

Packet reordering is an important metric for voice because packets have to be played back sequentially at the receiver in the same order that they have been sent. Thus, packets arriving earlier than their predecessors have to be buffered for reordering. We measure it according to the recommendation from [24], which is based on

the monotonic ordering of sequence numbers with a constant increment (denoted by Seq_{inc}). Specifically, in CPN packets are identified successively in the sending sequence with increment of '1'. To detect packet reordering, at the receiver we reproduce the sender's identifier function where the variable $NextExp$ is used to represent the Next Expected Identifier, which is incremented by Seq_{inc} once the in-order packet arrives. Given S is the identifier of the current arrival, if $S < NextExp$, the packet is reordered, else update $NextExp \leftarrow S + Seq_{\text{inc}}$.

To quantify the degree of desequencing, we also defined the 'Packet Reordering Ratio/Rate' and the 'Packet Reordering Density' denoted by $Density_r$, so that we may differentiate between isolated and bursty packet reordering as well as to measure the degree of burstiness of packet reordering, which may affect the packet drop rate of the resequencing buffer at the receiver. $Density_r$ is calculated as:

$$Density_r += \begin{cases} Cout_r^2 & \text{for bursty packet reordering} \\ Cout_r & \text{for isolated packet reordering.} \end{cases} \tag{4}$$

where $Cout_r$ is the number of successively reordered packets; it resets to zero when the in-order packets arrive and is incremented when reordering occurs.

The recommendation in [25] states that packet loss should be reported 'seperately on packets lost in a network, and those that have been received but then discarded by the jitter buffer' at the receiver for real-time packet delivery, because both have an equal effect on the quality of voice services, which is also denoted as packet end-to-end loss.

Packet loss within the network is detected for a packet that is sent out but not received by its destination node based on the matching of the packet identifier, the source and destination IP address. The resequencing buffer at the receiver is necessarily of finite length so that packets arriving to a buffer that is full will be discarded, and packets will have to be forwarded after a given time-out even when their predecessors have not arrived in order to avoid excessive time gaps with their predecessors that have already been played back. Thus, packets that arrive later than the expected playback time will also be discarded. As we cannot get directly access to the run-time version of the VOIP application, we had to simulate the operation of a jitter buffer, which employs resequecing so as to study packet discards and the buffer queue length, and their correlation with packet reordering and packet loss. We also defined the 'Packet (end-to-end) Loss Ratio/Rate/Density' consistent with (4).

4 Experimental Results

Our experiments were carried out on a wired test-bed network consisting of eight nodes with the topology shown in Fig. 2, whereby multiple paths are available for packets delivery between arbitrary source-destination pair. CPN was implemented as a loadable kernel module [19] running under linux 2.6.32 at each node. Adjacent nodes are connected with 100 Mbps Ethernet links.

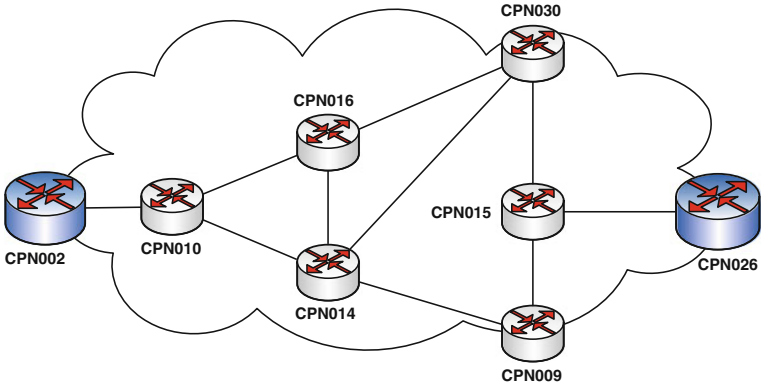


Fig. 2 CPN testbed network topology in the experiment

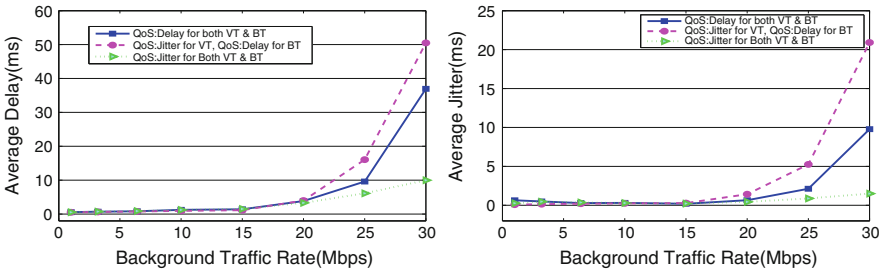


Fig. 3 The performance for voice traffic under varied background traffic conditions

We installed 'Linphone', a VOIP phone, at each node in the network testbed to generate actual voice traffic, while UDP traffic generation applications are running to introduce background traffic flows with a range of data rates to vary load conditions. As human listeners of voice are sensitive to the time-based QoS metrics 'delay' and 'delay variation', our experiments were conducted with voice traffic and one of the two QoS requirements, in the simultaneous presence of several background traffic flows with the same or the other QoS goal for the duration of ten minutes. We repeated each experiment with data rates of 1M, 2M, 3.2M, 6.4M, 10M, 15M, 20M, 25M, 30M bps and packet size of 1024 bytes for background traffic. Measurements of the voice traffic flow were gathered between CPN002 and CPN026 as this source-destination pair has the most intermediate nodes.

The measurements shown in Fig. 3 indicate that when we use Jitter Minimisation as the QoS goal both for the voice itself and the background traffic, jitter appears to be indeed minimised, but also delay and traffic loss are reduced for the voice traffic, because path switching is also reduced, alleviating route oscillations at heavy traffic loads. From observations for the voice traffic flow between CPN002 and CPN026 during the test interval considered while increasing the rate of the background traffic gradually, as shown in Fig. 4 (Left) as the rate of background traffic reached 20 Mbps,

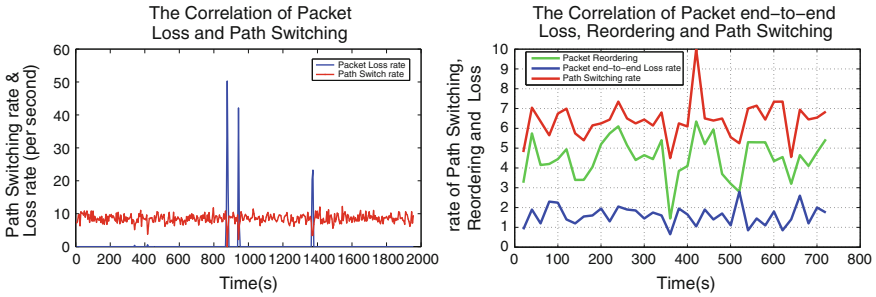


Fig. 4 The correlation of packet loss and path switching under medium (L) and heavy (R) traffic condition

it is seen that in the intervals (800–900 s, 900–1,000 s, 1,300–1,400 s), bursty packet loss occurred when path switching rates were low. This seems to arise from the fact that when a given path used by voice traffic satisfies the QoS criterion for a long time, and the path switching rate is close to '0', this path attracts other traffic, becoming saturated with packet loss ratio reaching '1'. Subsequently, the performance degradation is detected by the SPs and they move the traffic to less loaded paths.

As the rate of the background traffic increased to 30 Mbps, We can see from Fig. 4 (Right) that the occurrence of packet desequencing was frequent and varied proportionally with packet path switching, which demonstrates that packet reordering is mainly due to path switching in CPN. It is not easy to observe the correlation of packet path switching and packet loss from the figure. It is possibly because under heavy traffic conditions, packet loss is not only due to link saturated, route oscillation induced by heavy traffic loads, but also leads to the occurrence of loss. We can also find that an adequate path switching rate is beneficial to loss reduction, but if path switching rate is increased excessively, it is converted to route oscillation, which also lead to packet loss. To evaluate the packet drops at the receiver, we also used the voice packets received at CPN026 with the background traffic at rate of 30 Mbps as the input data to the simulation. It was found that the bursty packet loss and reordering that can be provoked by path switching within a network, i.e. the successive occurrence of packet loss and reordering, will induce delays for other packets in the output resequencing buffers of the VOIP codec, which in turn can provoke buffer overflow and further losses.

References

1. H. Toral-Cruz, J. Argaez-Xool, L. Estrada-Vargas, D. Torres-Roman, In Technology pp. 79–94 (2011)
2. FCC, in Office of Engineering and Technology and Consumer and Governmental Affairs Bureau (Federal Communications Commission, Washington, DC, USA, 2013)
3. C. Dovroli, K. Gummadi, A. Kuzmanovic, S.D. Meinrath, in *SIGCOMM CCR* (ACM, 2010)

4. S. Basso, M. Meo, A. Servetti, J.C.D. Martin, in *ACM SIGCOMM W-MUST* (ACM, 2012)
5. C. Diot, F. Gagnon, Comput. Netw. **31**(5), 475 (1999)
6. B. Zhang, R. Liu, D. Massey, L. Zhang, ACM SIGCOMM Computer Communication Review (CCR) special issue on Internet Vital Statistics (2005)
7. E. Katz-Bassett, C. Scott, D. Choffnes, I. Cunha, V. Valancius, N. Feamster, H. Madhyastha, T. Anderson, A. Krishnamurthy, in *ACM SIGCOMM* (CM, 2012)
8. E. Gelenbe, Acta Informatica **12**, 285 (1979)
9. E. Gelenbe, R. Lent, A. Montuori, Z. Xu, *Proceedings of the International Conference on Performance and QoS of Next Generation Networking* (Nagoya, Japan, 2002), pp. 3–17 (Opening Invited Paper)
10. E. Gelenbe, N. Schmajuk, J. Staddon, J. Reif, Robot. Auton. Syst. **22**(1), 23 (1997)
11. E. Gelenbe, M. Gellman, R. Lent, P. Liu, P. Su, in *Proceedings of the First International Conference on Autonomic Computing* (New York, NY, 2004), pp. 232–239. http://csdl.computer.org/comp/proceedings/icac/2004/2114/00/21140232abs.htm
12. A. Filippoupolitis, E. Gelenbe, in *Human System Interactions*, HSI'09. 2nd Conference on (2009), pp. 323–330
13. N. Dimakis, A. Filippoupolitis, E. Gelenbe, Comput. J. **53**(9), 1384 (2010)
14. E. Gelenbe, T. Mahmoodi, in *International Conference on Smart Grids, Green Communications, and IT Energy-aware Technologies(Energy 2011)* (Venice, Italy, Paper No. Energy-2011-1-20-50090, 2011)
15. E. Gelenbe, C. Morfopoulou, Comput. J. **54**(6), 850 (2011)
16. E. Gelenbe, Neural Computation 5(1), 154 (1993). doi:10.1162/neco.1993.5.1.154
17. E. Gelenbe, R. Lent, Z. Xu, J. Comput. Netw. **37**(6), 691 (2001)
18. E. Gelenbe, R. Lent, A. Nunez, Proc. IEEE **92**(9), 1478 (2004)
19. E. Gelenbe, Comm. ACM **52**(7), 66 (2009)
20. E. Gelenbe, R. Lent, A. Montuori, Z. Xu, in *Proceedings of the IEEE MASCOTS Conference (Ft. Worth, TX, 2002)*, pp. 3–12 (Opening Keynote Paper)
21. E. Gelenbe, P. Liu, in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks* pp. 517–521 (2005)
22. F. Baccelli, E. Gelenbe, B. Plateau, J. ACM **31**(3), 474 (1984)
23. M. Gellman, Qos routing for real-time traffic. Ph.D. thesis, Imperial College London (2007)
24. A. Morton, L. Ciavattone, G. Ramachandran, J. Perser, Request for Comments, 2006
25. T. Friedman, R. Caceres, A. Clark, Request for Comments, 2003

A Cooperative Emergency Navigation Framework Using Mobile Cloud Computing

Huibo Bi and Erol Gelenbe

Abstract The use of wireless sensor networks (WSNs) for emergency navigation systems suffer disadvantages such as limited computing capacity, restricted battery power and high likelihood of malfunction due to the harsh physical environment. By making use of the powerful sensing ability of smart phones, this paper presents a cloud-enabled emergency navigation framework to guide evacuees in a coordinated manner and improve the reliability and resilience in both communication and localization. By using social potential fields (SPF), evacuees form clusters during an evacuation process and are directed to egresses with the aid of a Cognitive Packet Networks (CPN)-based algorithm. Rather than just rely on the conventional telecommunications infrastructures, we suggest an Ad hoc Cognitive Packet Network (AHCPN)-based protocol to prolong the life time of smart phones, that adaptively searches optimal communication routes between portable devices and the egress node that provides access to a cloud server with respect to the remaining battery power of smart phones and the time latency.

Keywords Coordinated emergency navigation · Infrastructure-less · Cloud computing · Ad hoc Cognitive Packet Networks

1 Introduction

With the rapid development of sensor technology, emergency evacuation in built environments requires time-critical response with respect to multi-domain sensing, data interpreting and information transmitting. Wireless sensor network-based emergency management systems cannot provide optimal solutions due to their limited computing capability, battery power and storage capacity. Although the evolution from homogeneous architecture with functionality-identical sensors to heterogeneous architectures with separated sensing and decision subsystems makes WSNs-based systems

H. Bi (✉) · E. Gelenbe (✉)
Department of Electrical and Electronic Engineering,
Intelligent Systems and Networks Group,
Imperial College, London, England
e-mail: huibo.bi12@imperial.ac.uk

E. Gelenbe
e-mail: e.gelenbe@imperial.ac.uk

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_5

more energy-efficient and fault-tolerant, most decision supporting subsystems that consist of lightweight decision nodes still suffer from resource restriction problems.

Owing to the high processing power and low risk level, cloud-computing has become a dominating technology and tends to revolutionise the emergency management landscape. Accompanying this tendency, a new interest has been aroused to consider smart phones as simple clients for the back-end Cloud due to their sensing ability and popularity [1]. For example, Ref. [2] presents an emergency navigation system based on smart phones and active temperature RFID sensor tags to calculate evacuation routes and intensive computations are offloaded to a cloud server. Due to the limitation of indoor positioning and unavailability of GPS, in [3] a smart phone assisted system to locate evacuees with pedometry-based localization and image-based positioning is suggested. The cloud-based server can obtain the position of individuals by matching the image snapshots uploaded by evacuees and then provide uncrowded routes for users. However, current cloud-enabled emergency response systems with the aid of portable devices do not consider the impact of the significant energy consumption in the client side during the communication process. On the other hand, smart phones using the UMTS protocol can easily come under attack [4, 5] including various forms of denial of service attacks [6].

Much research has demonstrated cooperative strategies have a significant positive influence on multi-agent systems and related studies have emerged for decades as organizational paradigms in agent-based structures [7] and autonomous search by large scale robotic systems [8–10]. However, most research on emergency evacuation focuses on developing different models [11] to simulate the crowd or coordinated behaviours such as kin behaviour [12]. Although previous work [13] has proposed a resilient emergency support system (ESS) with the aid of opportunistic communications [14] to study the impact of "passive" cooperation among evacuees which exchange emergent messages with other civilians within the communication range, no active mechanism is used to improve the information dissemination efficiency.

Thus, in this paper we propose a cloud-based emergency navigation system, which uses a cooperative strategy to egress evacuees in loose clusters, and we also present a power-aware quality of service (QoS) metric to prolong the life time of smart phones used in an evacuation to connect to the access point(s) of the cloud system that carries out intensive computations.

The remainder of the paper is organised as follows: In Sect. 2, we recall the concept of Cognitive Packet Networks, and then describe related algorithms in Sect. 3. The simulation model and assumptions are introduced in Sect. 4 and results are presented in Sect. 5. Finally, we draw conclusions in Sect. 6.

2 Cognitive Packet Network

The Cognitive Packet Network (CPN) introduced in [15–17] was originally proposed for large-scale and fast-changing packet networks. By employing cognitive or smart packets (SPs), CPN can discover optimal routes rapidly and heuristically and realise

continuous self-improvement. Contrary to conventional routing protocols, in the CPN, intelligence is realised by using Random Neural Networks [18–23] and is constructed into smart packets other than protocols. Hence, cognitive packets can discover optimised routes with their predefined goals and improve QoS by learning from their own investigations and experience from other packets.

The AHCPN [24] is a variant of the original CPN. It can replace most of broadcast transmissions with unicast transmissions and adapt to highly dynamic Ad hoc network environments rapidly.

3 The Cloud-Enabled Emergency Navigation Framework

The framework contains a user layer which is composed of evacuees with portable devices and a Cloud layer which performs intensive computations. In the following subsections, we concentrate on the coordinated emergency navigation algorithm and the energy efficient protocol due to the space constraints.

3.1 Coordinated Emergency Navigation

Evacuating in a cooperative manner can increase the probability for evacuees to obtain assistance when at risk and reduce energy utilization of smart phones by relaying data. In this section, we leverage the SPF to cluster evacuees into groups during an evacuation. The initial SPF algorithm [25] consists of global control forces and local control forces. The global control forces coordinate the individuals and determine the distribution of the individuals while the local control forces dominate the personal behaviours. In our treatment, we replace the local control forces with the CPN-based algorithm to control the individual behaviour of evacuees and use the global forces to regulate intra-group behaviours. Details of the CPN-based emergency evacuation algorithm can be seen in [26] and [27]. The force between two evacuees can be calculated from the following equation:

$$f(r) = -\frac{c_1}{r^{\sigma_1}} + \frac{c_2}{r^{\sigma_2}} \tag{1}$$

where c_1, c_2, σ_1 and σ_2 are constants and r is the distance between two evacuees. Term $-\frac{c_1}{r^{\sigma_1}}$ presents the repulsive force while $\frac{c_2}{r^{\sigma_2}}$ depicts the attractive force. We assume that each civilian can only be affected by other civilians within 20 m. If the distance between two civilians is smaller than 7 m, a repulsive force will be generated, otherwise, a attractive force will be produced. To achieve this, we set c_1 to 20, c_2 to 15, σ_1 to 0.9478 and σ_2 to 0.8, respectively. This ensures the civilians to evacuate in loose groups without increasing the level of congestion.

To combine the SPF and the CPN-based algorithm, we use a simple scheme to randomly choose either the decision of the SPF or the CPN as the next decision at a time. When a SPF related decision is chosen, we adopt the associated neighbour node, which has the most matched direction with the resultant force as the next hop.

3.2 Power-Aware Protocol

As the front-end component of the cloud-enabled framework, the operational time of smart phones becomes a bottleneck of the system due to the large amount of energy consumption during information exchanges. Because the remaining battery power of the portable devices yields normal distribution when an emergency event occurs and the power consumption of different communication modes varies, it is not a optimal strategy for each smart phone to exchange information with the Cloud through 3G directly. To realise energy efficiency and maximise the average battery lifetime of smart phones, we employ an AHCPN-based algorithm to relay sensory data before ultimately uploading to the Cloud. The AHCPN-based algorithm is deployed on the smart phones to search power-saving paths to convey sensory data to the Cloud. The QoS criterion we used is inspired by the energy aware metric in [24]. Here we also employ the "path availability" notion and construct the metric as follows.

$$G_{ed} = \alpha \prod_{i=0}^{n-1} P_a(\pi(i), \pi(i+1)) \left\{ \sum_{i=0}^{n-1} D(\pi(i), \pi(i+1)) \right\} \tag{2}$$

where π represents a particular path, n is the number of nodes (smart phones) on the path π, and $\pi(i)$ is the ith node on the path π. Term $P_a(\pi(i), \pi(i+1))$ is the availability of the edge between $\pi(i)$ and $\pi(i+1)$. $D(\pi(i), \pi(i+1))$ is the delay cost for a packet to transmit from $\pi(i)$ to $\pi(i+1)$. Term α is a constant that coordinates the relation between the path availability and delay.

Path availability is affected by the remaining battery power of a smart phone and the estimated power consumption of transmitting a piece of information:

$$P_a(\pi(i), \pi(i+1)) = \frac{B_i^C}{B_i^C - B_i^U} \tag{3}$$

where B_i^C represents the current remaining battery power of a node (smart phone) $\pi(i)$ and B_i^U depicts the power utilisation at node $\pi(i)$ to convey a certain piece of information. If the potential power utilisation is larger than the remaining battery, this node will be excluded.

4 Simulation Model and Assumptions

Since mathematical models [28] cannot handle the full complexity that is encountered during an emergency, the performance of the proposed algorithms is evaluated in fire-related scenarios based on the Distributed Building Evacuation Simulator (DBES) [29, 30]. The building model is a three-storey canary wharf shopping mall as shown in Fig. 1. Initially, evacuees are randomly scattered in the building and are equipped with smart phones with random remaining battery power. When the battery power of a smart phone is depleted, the evacuee will wander or follow other evacuees in the line of sight to exits.

We assume that no related sensors are pre-installed in the built environment. Hence, hazard information and location of civilians will be collected by mobile phones with built-in cameras and analysed in the Cloud. We also hypothesis that indoor communication infrastructures such as Wi-Fi access points are unavailable but a few cloud access points can be rapidly deployed between the emergency location and the Cloud when a hazard occurs. According to literature [31–34], the energy consumption model of an individual smart phone is given in Table 1.

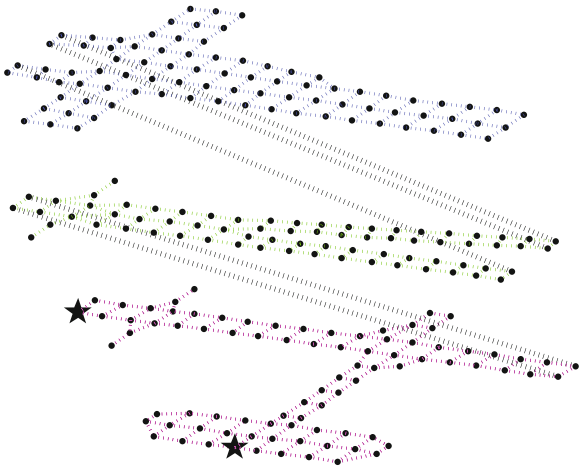


Fig. 1 Graph representation of the building model: vertices are positions with landmarks that evacuees can easily identify their locations by uploading snapshots and matching them with pre-known landmarks which are stored in cloud severs, edges are physical links that evacuees can move inside the building. The two *black* stars on the ground floor mark the position of the building's exits

Table 1 The energy consumption of a smart phone in Joule with regard to the transferred data x in byte and the related signal rate

Communication model	Download	Upload	Signal rate
3G	$0.001224x$	$0.0003375x$	2 Mb/s
Bluetooth	$0.0001377x$	$0.00012012x$	1 Mb/s

5 Results and Discussion

To evaluate the navigation algorithm which combines CPN and SPF (CPN&SPF) as well as the AHCPN-based energy efficient protocol, we design an experiment which involves four scenarios with 30, 60, 90, 120 evacuees, respectively. The Dijkstra's shortest path algorithm and CPN-based algorithm with time metric (CPNST) [35] are performed for comparison purpose.

Figure 2 indicates that both CPNST and CPN&SPF achieve more survivors than Dijkstra's algorithm due to the congestion-ease mechanisms in both CPNST and SPF. Compared with CPNST, CPN&SPF performs slightly better in densely-populated scenarios because CPN&SPF organises evacuees as loose clusters and is easier to balance the power utilisation among smart phones. As a result, the probability of being trapped due to the depletion of smart phones is decreased. Furthermore, the coordinated behaviour increases the probability for an evacuee with depleted smart phone to follow other evacuees rather than wander in the building.

As can be seen from Fig. 3, Dijkstra's algorithm achieves least number of drained smart phones in 3G mode. This is because both the CPNST and CPN&SPF do not follow the shortest path. Hence, evacuees will traverse more landmarks and upload more photos to the Cloud. However, by employing the AHCPN protocol, the number of depleted smart phones decreases significantly for all three algorithms. Moreover, there is no drained smart phone when combining the CPN&SPF algorithm with the AHCPN protocol. This confirms that routing evacuees in loose groups can contribute to the power-balancing of smart phones. The results also indicates that although AHCPN may consume extra energy because of sending smart packets periodically, it can effectively reduce the number of drained smart phones by balancing the remaining battery power of smart handsets.

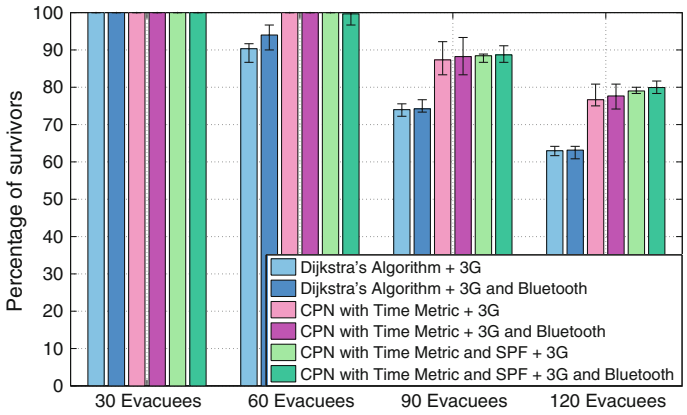


Fig. 2 Percentage of survivors for each scenario. The results are the average of five randomised simulation runs, and error bars shows the min/max result in any of the five simulation runs

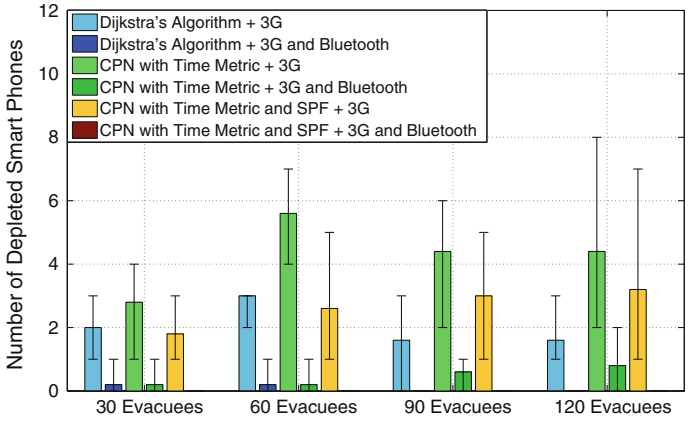


Fig. 3 The number of drained smart phones of five iterations. The error bars represent the min/max values found in the five simulations

6 Conclusions

In this paper, we propose an infrastructure-less indoor emergency response system to evacuate civilians with the aid of smart handsets and cloud servers. A coordinated emergency navigation algorithm is proposed to guide evacuees in loose groups. The experimental results prove that the algorithm can increase the survival rate by reducing the number of drained smart phones in an evacuation process and raising the likelihood for an evacuee with a depleted mobile device to encounter other evacuees in the line of sight and follow them to egresses. Due to the considerable energy consumption between the Cloud and the smart phones during communication processes, an AHCPN-based energy efficient protocol is also presented to prolong the life time of smart handsets. Simulations indicate that the protocol can significantly decrease the number of drained smart phones in an evacuation process and balance the remaining battery power among portable devices.

References

1. E. Gelenbe, F.J. Wu, Future Internet **5**(3), 336 (2013)
2. L. Chu, S.J. Wu, in *Nano, Information Technology and Reliability (NASNIT), 2011 15th North-East Asia Symposium on* (IEEE, 2011), pp. 45–48
3. J. Ahn, R. Han, in *Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific* (IEEE, 2011), pp. 70–77
4. E. Gelenbe, G. Görbil, D. Tzovaras, S. Liebergeld, D. Garcia, M. Baltatu, G.L. Lyberopoulos, in *ISCIS, Lecture Notes in Electrical Engineering*, ed. by E. Gelenbe, R. Lent. Lecture Notes in Electrical Engineering, vol. 264 (Springer, 2013), pp. 369–378
5. E. Gelenbe, G. Gorbil, D. Tzovaras, S. Liebergeld, D. Garcia, M. Baltatu, G. Lyberopoulos, in *Proceedings of the 2013 IEEE Global High Tech Congress on Electronics (GHTCE'13)* (2013)

6. O.H. Abdelrahman, E. Gelenbe, in *Proceedings of IEEE International Conference on Communications, ICC 2014*, Sydney (2014)
7. B. Horling, V. Lesser, Knowl. Eng. Rev. **19**(4), 281 (2004)
8. E. Gelenbe, N. Schmajuk, J. Staddon, J. Reif, Robotics Auton. Syst. **22**(1), 23 (1997)
9. Y. Cao, E. Gelenbe, in *Proceedings SPIE 3079, Detection and Remediation Technologies for Mines and Minelike Targets II* (1997), p. 691. doi:10.1117/12.280898
10. Y. Cao, E. Gelenbe, Eur. J. Oper. Res. **108**(2), 319 (1998)
11. X. Zheng, T. Zhong, M. Liu, Building Environ. **44**(3), 437 (2009)
12. L. Yang, D. Zhao, J. Li, T. Fang, Building Environ. **40**(3), 411 (2005)
13. G. Gorbil, E. Gelenbe, in *Computer and Information Sciences III* (Springer, 2013), pp. 249–257
14. L. Pelusi, A. Passarella, M. Conti, Communications magazine. IEEE **44**(11), 134 (2006)
15. E. Gelenbe, R. Lent, Z. Xu, *Performance and QoS of Next Generation Networking* (Springer, London, 2001)
16. E. Gelenbe, R. Lent, A. Nunez, Proc. IEEE **92**(9), 1478 (2004)
17. E. Gelenbe, Commun. ACM **52**(7), 66 (2009)
18. E. Gelenbe, Neural Comput. **1**(4), 502 (1989)
19. E. Gelenbe, Neural Comput. **2**(2), 239 (1990)
20. E. Gelenbe, Neural Comput. **5**(1), 154 (1993)
21. E. Gelenbe, S. Timotheou, Neural Comput. **20**(9), 2308 (2008)
22. E. Gelenbe, E. Seref, Z. Xu, Proc. IEEE **89**(2), 148 (2001)
23. E. Gelenbe, Comput. J. **55**(7), 848 (2012)
24. E. Gelenbe, R. Lent, Ad Hoc Netw. **2**(3), 205 (2004)
25. J.H. Reif, H. Wang, Robotics Auton. Syst. **27**(3), 171 (1999)
26. H. Bi, A. Desmet, E. Gelenbe, in *Proceedings of the 28th International Symposium on Computer and Information Sciences (ISCIS'13)* (Springer, London, 2013)
27. H. Bi, Future Internet **6**(2), 203 (2014)
28. E. Gelenbe, Acta Inf. **12**, 285 (1979)
29. A. Filippoupolitis, E. Gelenbe, in *Human System Interactions, 2009. HSI'09. 2nd Conference on* (2009), pp. 323–330
30. N. Dimakis, A. Filippoupolitis, E. Gelenbe, Comput. J. **53**(9), 1384 (2010)
31. A. Carroll, G. Heiser, in *Proceedings of the 2010 USENIX conference on USENIX annual technical conference* (2010), pp. 21–21
32. C. Haas, J. Wilke, V. Stöhr, in *Wireless Sensor Networks* (Springer, 2012), pp. 82–97
33. N. Balasubramanian, A. Balasubramanian, A. Venkataramani, in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference* (ACM, 2009), pp. 280–293
34. G. Kalic, I. Bojic, M. Kusek, in *MIPRO, 2012 Proceedings of the 35th International Convention* (IEEE, 2012), pp. 754–759
35. H. Bi, E. Gelenbe, in *Proceedings of the 4th International Workshop on Pervasive Networks for Emergency Management (PerNEM'14)* (IEEE, 2014), pp. 1–6

Part II
Data and Image Analytics

Integer Linear Programming Solution for the Multiple Query Optimization Problem

Tansel Dokeroglu, Murat Ali Bayır and Ahmet Cosar

Abstract Multiple Query Optimization (MQO) is a technique for processing a batch of queries in such a way that shared tasks in these queries are executed only once, resulting in significant savings in the total evaluation. The first phase of MQO requires producing alternative query execution plans so that the shared tasks between queries are identified and maximized. The second phase of MQO is an optimization problem where the goal is selecting exactly one of the alternative plans for each query to minimize the total execution cost of all queries. A-star, branch-and-bound, dynamic programming (DP), and genetic algorithm (GA) solutions for MQO have been given in the literature. However, the performance of optimal algorithms, A-star and DP, is not sufficient for solving large MQO problems involving large number of queries. In this study, we propose an Integer Linear Programming (ILP) formulation to solve the MQO problem exactly for a large number of queries and evaluate its performance. Our results show that ILP outperforms the existing A-star algorithm.

Keywords Linear programming · Multiple query optimization · A-star

1 Introduction

Multiple Query Optimization (MQO) is a research problem that has been studied since 1980s [1, 9, 13, 19]. Its goal is to minimize the total execution time of a set of queries by evaluating the common expressions only once. Query optimizers usually consider one query at a time and find an optimum solution by considering different join orders and alternative access paths to the tables. Multiple Query Optimization

T. Dokeroglu (✉) · M.A. Bayır · A. Cosar
Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
e-mail: tansel@ceng.metu.edu.tr

M.A. Bayır
e-mail: mbayir@microsoft.com

A. Cosar
e-mail: cosar@ceng.metu.edu.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_6

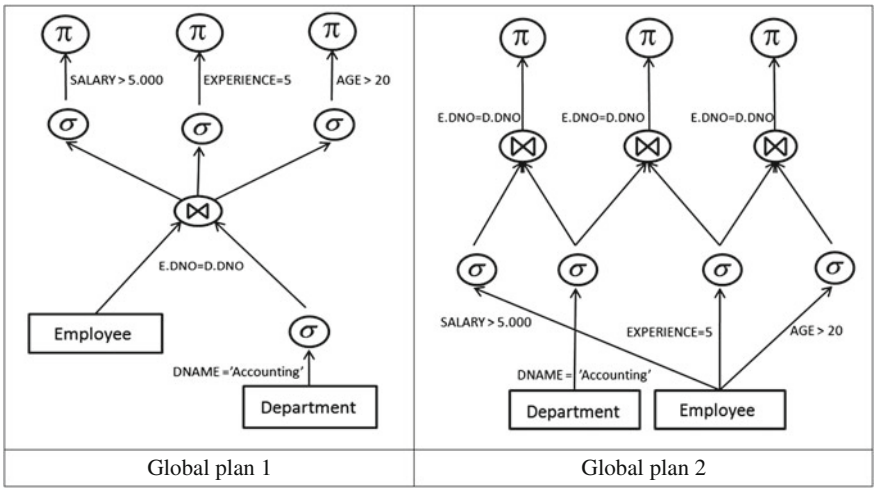


Fig. 1 Two global plans for processing queries Q_1, Q_2, and Q_3

executes many queries together with a global plan that uses has more common tasks as possible. Processing queries individually is not an efficient way when there exists a high number of common expressions that result in multiple redundant evaluations. Multiple Query Optimization detects common tasks among queries and creates a global query execution plan in which these tasks are evaluated only once. Consider the following example:

Q_1: Find the names of employees older than 20 and working in the accounting department.
Q_2: Find the names of employees having work experience of 5 years or more and currently working in the accounting department.
Q_3: Find the names of employees earning more than 5,000$ and working in the accounting department.

The 'Accounting' department is common to all three queries. Therefore, the selection operation for DName = "Accounting" can be shared by all of the above queries. In order to achieve even more sharing the selection operators, Salary $>5{,}000$, Experience $= 5$, and Age >20 can be delayed, and it becomes possible to share the result of DName = 'Accounting' selection operation. Two possible alternative plans for these queries are shown in Fig. 1.

When using single query optimization in relational databases, the query optimizer always produces plans that perform select operations as early as possible. However, in MQO it may be more profitable to delay some select operations unlike the conventional single query optimization because the total cost of executing all queries by using shared tasks only once may be cheaper than executing optimal query evaluation plans for each query separately.

A-star and Dynamic Programming (DP) algorithms have been developed for finding optimal solutions to MQO problem instances [20, 22]. However, the amount of memory required by A-star increases rapidly and large MQO problem instances cannot be solved using this technique. The DP formulation is very conservative in its memory usage but its run time performance is almost the same as that of A-star. Integer Linear Programming (ILP) is a very popular technique that is widely used for finding optimal solutions of optimization problems and its performance has been very good for many problem domains. Formulation of MQO using ILP is difficult because the objective function is based on the set of all tasks in selected plans while the solution to the MQO is in terms of the selected alternative plans. In this work, we solve this conflict by defining additional constraints:

- At least one of the alternative plans of each query must be selected,
- All tasks of a selected plan must also be selected (i.e. its corresponding variable is assigned integer value 1), which guarantee a valid solution.

2 Related Work

MQO is an NP-hard problem [8–10, 17–19]. A detailed theoretical study of MQO can be found in [7]. The MQO problem consists of mainly two stages. In the first stage, common tasks of the queries are detected. In [4], detailed information about the detection techniques is given and a multigraph for representing and facilitating the processing of multiple queries is proposed. A connection graph which is an extension of a query graph is used in [21]. Second stage of the MQO problem schedules the query plans to provide the minimum total execution time for a query batch. A state space search formulation is defined in [19] with an A-star algorithm using bounding functions and intelligent state space expansion. Several heuristics and cost prediction schemes are used to prune states and reduce the memory usage of A-star algorithm in [5]. It has been shown that A-star and DP can be used for optimally solving the MQO problem for up to twenty queries in practical times [3, 16, 18]. In [16], state-of-the-art MQO techniques of using joint expressions are applied for the execution of multiple related queries and efficient maintenance of materialized views. An efficient relational database integration system is proposed to provide a collection of virtual views for integrating data objects from various distributed, heterogeneous, and autonomous data sources [11].

Our main contribution in this study is to formulate the MQO problem with ILP and show that it solves MQO problem in much less time than the best available optimal solution algorithm, A-star (for large MQO problems containing from 20 up to 100 queries). It is also possible to find sub-optimal solutions for such large MQO problems using Genetic Algorithms (GA) as proposed in [2] and both the quality of the solutions are good and the runtime of GA is linear in terms of MQO problem size. However, GAs do not always guarantee the optimal solution for large problem sizes.

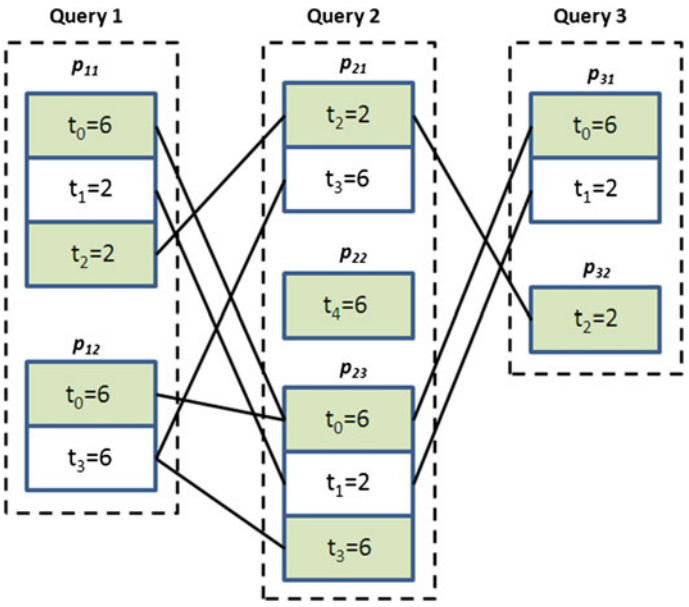


Fig. 2 Sample MQO problem with 3 queries, 7 plans, and 5 tasks

3 ILP Model for MQO

The MQO problem consists of a set of queries (Q_1 through Q_n). Each query Q_i has a number of alternative plans each containing one or more tasks. An answer to a query is produced by executing all of the tasks in any one of the alternative plans of that query [19]. Tasks have positive integer cost values and a task may exist in one or more plans that could belong to any of the queries in the query batch. A solution to the MQO problem can be obtained by selecting exactly one of the alternative plans for each query (see Fig. 2). There are three queries, Q_1, Q_2, and Q_3. A total of five distinct tasks exist in all of the alternative plans. The ith task is represented by t_i . In our example, the existing tasks are t_0, t_1, t_2, t_3, and t_4 that have corresponding costs of $t_0 = 6, t_1 = 2, t_2 = 2, t_3 = 6$, and $t_4 = 6$. Q_1 and Q_3 have two alternative plans each, and Q_2 has three plans, so there are seven plans in total. Each plan is represented as p_{ij} which means that it is the jth plan for ith query. The plans for the first query are $p_{11} = \{t_0, t_1, t_2\}$ and $p_{12} = \{t_0, t_3\}$. The plans for the second query are $p_{21} = \{t_2, t_3\}$, $p_{22} = \{t_4\}$, and $p_{23} = \{t_0, t_1, t_3\}$. The plans for the third query are $p_{31} = \{t_0, t_1\}$ and $p_{32} = \{t_2\}$.

Many real life problems have been solved with ILP models until now [14]. There are also some studies in the field of database. In [12], the problem of allocating relational database fragments to a network to minimize the overall communication cost for processing a given set of transactions is solved with ILP. An efficient *index selection* ILP model is proposed in [15]. However, there is no ILP solution model

proposed for MQO problem such as our technique. The general form of the ILP is as follows:

$$\text{Maximize } c^T x \tag{1}$$

$$\text{Subject to: } A\,x \leq b \tag{2}$$

$$x \in Z^n \tag{3}$$

Equation 1 is used as a vector of n integer variables (Eqs. 1–3). x is a vector of $(x_1, \ldots, x_n)^T$, A is an $m \times n$ matrix of rational data, c is a $1 \times n$ matrix, and b is an $m \times 1$ matrix. The MQO problem instance in Fig. 2 can be modeled with ILP as follows:

$$x_1 + x_2 = 1 \tag{4}$$

$$x_3 + x_4 + x_5 = 1 \tag{5}$$

$$x_6 + x_7 = 1 \tag{6}$$

$$-10x_1 + 6x_8 + 2x_9 + 2x_{10} \geq 0 \tag{7}$$

$$-12x_2 + 6x_8 + 6x_{11} \geq 0 \tag{8}$$

$$-8x_3 + 2x_{10} + 6x_{11} \geq 0 \tag{9}$$

$$-6x_4 + 6x_{12} \geq 0 \tag{10}$$

$$-14x_5 + 6x_8 + 2x_9 + 6x_{11} \geq 0 \tag{11}$$

$$-8x_6 + 6x_8 + 2x_9 \geq 0 \tag{12}$$

$$-2x_7 + 2x_{10} \geq 0 \tag{13}$$

The goal is to minimize the below objective function:

$$6x_8 + 2x_9 + 2x_{10} + 6x_{11} + 6x_{12} \tag{14}$$

Plans are represented by variables x_1 through x_7 and have values zero or one showing whether the corresponding plan is selected for inclusion in the global plan or not. Q_1 has plan variables x_1 and x_2, Q_2 has plan variables x_3, x_4, x_5, and finally Q_3 has plan variables x_6 and x_7. Equations 4–6 represent the constraints that only one plan can be chosen for each query. Each task is assigned a variable x_8 through x_{12} to represent whether that task is included in the global query plan. Equations 7–13 represent the constraints that if a plan is selected for a query, all of the tasks in that

plan must also be selected. For example, Eq. 7 represents query plan 1 (p_{11}) of Q_1, and tasks of p_{11} are t_0 (cost = 6), t_1 (cost = 2), t_2 (cost = 2) and are coded by variables x_8, x_9, and x_{10}, respectively. By multiplying each task cost by its variables we obtain $6x_8 + 2x_9 + 2x_{10}$ and this sum is subtracted from $10x_1$ (meaning that p_{11} is selected) that must be greater than or equal to zero (even though a plan is not selected, one of its tasks may exist in another selected plan). Thus, for each plan in the MQO problem instance we produce Eqs. (7–13).

Finally, the objective function (Eq. 14) must be minimized to determine the set of plans such that the total cost of all of the tasks in these plans is minimized.

4 A-Star Algorithm

There are several variants of A-star algorithm used for solving the MQO problem [5]. A-star algorithm starts with a *null* initial state (meaning that no queries have been assigned a plan yet). At each level of the tree, one of the alternative plans is chosen and assigned to a query with no assigned plan. Given an admissible cost estimation heuristic function; A-star algorithm is guaranteed to provide optimal results with a minimal number of expanded states. The number of expanded states in A-star search determines the memory usage and greatly affects the efficient and quick solution of MQO problem instances. The set of selected tasks, t_{sel}, in assigned plans for queries Q_1 to Q_k (where the total number of queries is denoted by Q and $k \leq Q$) is represented as: where NT_i is the number of tasks in the assigned plan for Q_i and PT_{ij} is the jth task in the assigned plan. The cost estimation of task t_i is defined as (if $t_i \notin t_{\text{sel}}$, it is zero otherwise):

$$\text{estCost}(t_i) = \frac{\text{realCost}(t_i)}{m_i} \tag{15}$$

In Eq. 15, m_i is the number of queries waiting for a plan to be assigned and has at least one alternative plan that contains task t_i. The estimated cost for plan p_{ij} is:

$$\text{estCost}(p_{ij}) = \sum\nolimits_{t_i \epsilon p_{ij}} \text{estCost}(t_i) \tag{16}$$

Let P_i be the number of alternative plans for q_i, and a plan has been assigned for queries Q_1,…, Q_k then the heuristic function used in our A-star algorithm is:

$$\begin{aligned} \text{h}(S_k) = & \sum_{t_x \epsilon t_{\text{sel}}} \text{realCost}(t_x) \\ & + \sum_{k \leq i \leq Q} \min\left(\text{estCost}(p_i, 1), \ldots, \text{estCost}(p_{i,p_i})\right) \end{aligned}$$

Six alternative query ordering methods have been defined to minimize the estimation error in the plan cost estimation function (17) and this query order remains constant throughout the search. The method of ordering the queries in decreasing average query cost has been experimentally shown to be the best performing one. This query ordering is used for the A-star implementation in our study as well.

5 Experimental Setup and Results

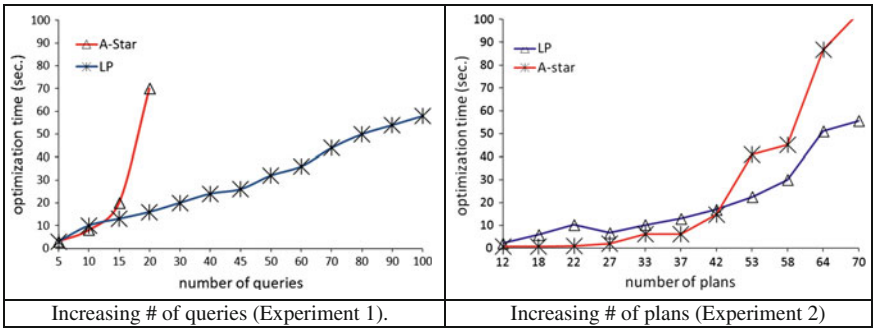
Python 2.7 is used to implement the A-star and ILP. The number of plans, tasks, and the execution time of the tasks used in the problem sets are randomly generated [2]. A heuristic value, *input size*, is used to estimate the search space. The *input size* is defined by the multiplication of the number of queries q, the average number of tasks per plan t_{av}, and the average number of plans per query p_{av}. We first verified that both A-star and ILP algorithms produce the same results. Later, we compared the time spent by ILP and A-star algorithms to find an optimal solution on the same MQO problem instances. The values used in four different problem sets with varying number of queries, plans, and tasks are given in Table 1.

For each problem set, one of the parameters is increased gradually. For example, in the first problem set the number of queries is varied between 5 and 100, each query has between 10 and 15 plans, and the number of tasks in each plan is randomly chosen between 9 and 11. The fourth problem set uses a moderate number of plans and tasks to show the capability of ILP to provide optimal results without having an exponential execution time while the A-star algorithm shows severe performance degradation. The results of the problem sets 1, 2, 3, and 4 are shown in Figs. 3, 4, and 5.

It is clearly seen from the results given in Fig. 3 that ILP outperforms the A-star algorithm for practical MQO problem sizes. Since the ILP formulates each task as a variable, the complexity of the ILP formulation increases in parallel with the number of tasks while this has less impact in the performance of A-star algorithm and both algorithms exhibit similar performance with A-star performing marginally better (see Fig. 4). A-star cannot solve MQO instances with more than 20 plans when number of alternative plans is above 10, while ILP performs remarkably well and can solve MQO instances with 20 or more queries, even when the number of queries becomes

Table 1 Parameters used in the experiments

Set	Input size	Number of queries	Number of plans	Number of tasks
1	[625- 12,500]	[5,100]	[10,15]	[9,11]
2	[625, 3,375]	5	[10,70]	[9,11]
3	[625, 1,250]	5	[10,15]	[9,21]
4	[1,375, 5,500]	[25,100]	[4,7]	[9–11]



Increasing # of queries (Experiment 1). Increasing # of plans (Experiment 2)

Fig. 3 Performances of ILP and A-star algorithms

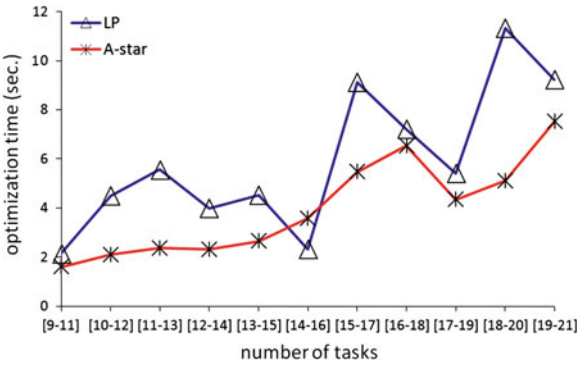


Fig. 4 Performances of ILP and A-star algorithms for increasing task count (Experiment 3)

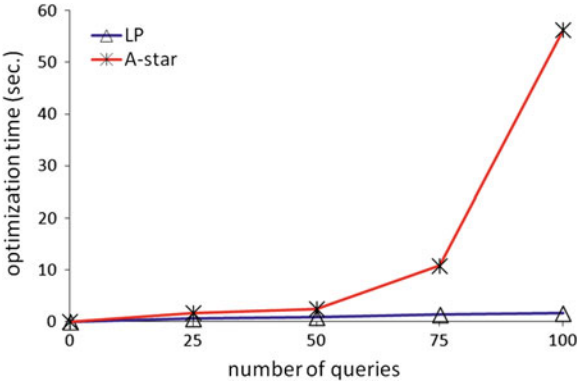


Fig. 5 Performances of ILP and A-star algorithms up to 100 queries (Experiment 4)

as high as 100. Figure 5 gives the results of the problem set 4 where each query has [4–7] plans and each plan has [9–11] tasks where A-star is able to find the solution but its execution times increases exponentially while ILP's running time increases almost linearly. A-star typically consumes large amounts of memory in these experiments which is a major disadvantage for using it in a query optimizer. The optimization time of ILP is much better than A-star algorithm when the number of queries is increased. By employing intelligent alternative plan generation techniques and keeping the number of tasks under tight control, it is guaranteed that the performance of ILP will be much better than A-star and will have almost linear running time in terms of the number of plans and queries. This makes ILP a very good candidate for inclusion in a multiple query optimizer.

6 Conclusions and Future Work

Multiple query optimization (MQO) is an NP-hard problem and many algorithms have been developed to solve this problem optimally. A-star heuristics have been reported to be one of the best performing algorithms in the literature. However, as the problem size grows, the difficulty of solving the MQO problem makes A-star algorithm prohibitively expensive. Until now, exact solutions of problem instances with 20 queries have been considered unsolvable. In this study, we propose an Integer Linear Programming (ILP) formulation for solving very large MQO problem instances with up to 100 queries, and experimentally show them to be solvable in seconds. To our knowledge, this is the first time that such large MQO problem instances have been solved optimally. We believe that further performance improvements can be achieved by using parallel ILP tools.

References

1. E. Angel, E. Bampis, L. Gourvès, On the minimum hitting set of bundles problem, *Algorithmic Aspects in Information and Management* (Springer, Berlin, 2008), pp. 3–14
2. M.A. Bayir, I.H. Toroslu, A. Cosar, Genetic algorithm for the multiple-query optimization problem. Syst. Man Cybern. Part C: Appl. Rev. IEEE Trans. **37**(1), 147–153 (2007)
3. J.R. Bernardino, P.S. Furtado, H.C. Madeira, Approximate query answering using data warehouse striping. J. Intell. Inf. Syst. **19**(2), 145–167 (2002)
4. F.C. Chen, M.H. Dunham, Common subexpression processing in multiple-query processing. Knowl. Data Eng. IEEE Trans. **10**(3), 493–499 (1998)
5. A. Cosar, E.P. Lim, J. Srivastava, Multiple query optimization with depth-first branch-and-bound and dynamic query ordering. in *Proceedings of the second international conference on Information and knowledge management* (ACM, 1993), (pp. 433–438)
6. A. Cosar, J. Srivastava, S. Shekhar, On the multiple pattern multiple object (MPMO) match problem. Army High Perform. Comput. Res. Cent. (1991)
7. A.A. Diwan, S. Sudarshan, D. Thomas, Scheduling and caching in multi-query optimization. In *Proceedings of 13th International Conference Management of Data* (2006)

8. S. Finkelstein, Common expression analysis in database applications. in *Proceedings of the 1982 ACM SIGMOD international conference on Management of data* (ACM, 1982), pp. 235–245
9. F. Polat, A. Cosar, R. Alhajj, Semantic information-based alternative plan generation for multiple query optimization. Inform. Sci. **137**(1), 103–133 (2001)
10. H.H. Lee, W.S. Lee, Adaptive two-level optimization for selection predicates of multiple continuous queries. J. Intell. Inform. Syst. **39**(2), 317–334 (2012)
11. R. Lee, M. Zhou, H. Liao, Request Window: an approach to improve throughput of RDBMS-based data integration system by utilizing data sharing across concurrent distributed queries. in *Proceedings of the 33rd international conference on VLDB* (2007), pp. 1219–1230
12. X. Lin, M. Orlowska, An Integer Linear Programming approach to data allocation with the minimum total communication cost in distributed database systems. Inform. Sci. **85**(1), 1–10 (1995)
13. G. Nan, M. Li, Energy-efficient query management scheme for a wireless sensor database system. EURASIP J. Wirel. Commun. Netw. (2010)
14. S.G. Nash, A. Sofer, *Linear and Nonlinear Programming*, vol. 692 (McGraw-Hill, New York, 1996)
15. S. Papadomanolakis, A. Ailamaki, An integer linear programming approach to database design. in *Data Engineering Workshop, 2007 IEEE 23rd International Conference on* (2007), pp. 442–449
16. J. Rao, K.A. Ross, Reusing invariants: a new strategy for correlated queries. in *ACM SIGMOD Record*, vol. 27, No. 2, (1998), pp. 37–48
17. A. Rosenthal, U.S. Chakravarthy, Anatomy of a Modular Multiple Query Optimizer. In *VLDB* (1988), pp. 230–239
18. P. Roy, S. Seshadri, S. Sudarshan, S. Bhobe, Efficient and extensible algorithms for multi query optimization. in *ACM SIGMOD Record*, vol. 29, No. 2, (2000), pp. 249–260
19. T.K. Sellis, Multiple-query optimization. ACM Trans. Database Syst. (TODS) **13**(1), 23–52 (1988)
20. I.H. Toroslu, A. Cosar, Dynamic programming solution for multiple query optimization problem. Inform. Process. Lett. **92**(3), 149–155 (2004)
21. E. Wong, K. Youssefi, Decomposition—a strategy for query processing. ACM Trans. Database Syst. (TODS) **1**(3), 223–241 (1976)
22. T. Dokeroglu, S.A. Sert, M.S. Cinar, Evolutionary multiobjective query workload optimization of Cloud data warehouses. Sci. World J. (2014)

A Graphical Model Approach for Multi-Label Classification

Meltem Cetiner and Yusuf Sinan Akgul

Abstract Multi-Label (ML) classification problem is the assignment of many labels to a given sample from a fixed label set. It is considered as the more general version of the Multi-Class (MC) classification problem and its practical application areas vary from medical diagnosis to paper keyword selection. The general structure of an ML classification system involves transforming the problem into simpler MC and Single-Class (SC) problems. One such method is the Binary Relevance (BR) method that treats each label assignment as an independent SC problem, which makes BR systems scalable, but not accurate for some cases. This paper addresses the label independence problem of BR by assuming the outputs of each SC classifiers as observation nodes of a graphical model. The final label assignments are obtained by standard powerful Bayesian inference from the unobservable node. The proposed system was tested on standard ML classification datasets that produced encouraging results.

Keywords Multi-label classification · Binary relevance · Graphical model · Label dependency

1 Introduction

Multi-Label (ML) classification is the process of assignment of zero or more labels to a given sample (or observation) represented by a vector of features. There are many practical ML classification examples in the real world. For example, the medical diagnosis process usually involves assignment of several different illness names to a patient represented by a features vector that includes the symptoms [9]. Another example is the assignment of keywords to papers where each paper can have different number of keywords [10]. Many other examples in the literature [6] can be found

M. Cetiner (✉) · Y.S. Akgul
Department of Computer Engineering, Gebze Institute of Technology, Gebze, Turkey
e-mail: mcetiner@bilmuh.gyte.edu.tr

Y.S. Akgul
e-mail: akgul@bilmuh.gyte.edu.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_7

such as the assignment of emotions to facial expressions, assignment of genre labels to movies, and assignment of functional categories to yeast genes. ML classification is usually compared with Multi-Class (MC) classification systems and Single-Class (SC) classification systems. Different from ML systems, for a given sample, MC assigns only a single class label among many classes. SC systems, on the other hand, can assign only a fixed single class label for each sample. The problem of ML classification is considered as a more general and difficult problem compared to MC and SC classification [3].

There are many ML systems proposed in the literature such as Nearest Neighbors (NN) [13], Bayesian Learners (BL) [2], Decision Trees (DT) [12], and Chain Classifiers (CC) [7]. The common problems that need to be addressed by these systems are twofold. First, the scalability problems need to be solved because the number of labels and the number of sample features can get very high for the ML problems. For example, the BL-based systems have to learn the relationships between the feature nodes for each label, which makes the overall system very complicated for some datasets with many features and labels [13]. The second problem of ML classification is the dependency between the labels, which is the main difference between the ML and MC systems. An MC classification system assumes that no two class labels can overlap on a given sample. However, the ML classification systems operate under the assumption that each sample can have many different labels, which creates the issue of dependence between the assigned labels for a given sample. For example, for the movie genre label assignment application, an assigned "Family" label for a movie is a very good indication for that movie not having a "Horror" label, which shows the inverse dependency between "Family" and "Horror" labels. The literature defines a base line ML classification algorithm, named Binary Relevance (BR), which is both simple and can perform very well for some tasks. BR defines a separate SC classifier for each label that decides if a label is relevant for a given sample. BR makes a one-versus-all decision for each label for the given sample. In other words, for a given sample, BR makes as many decisions as the number of labels or classes. This makes BR very scalable because the number of decision operations grow linearly with the number of labels. As a result, BR is used for many ML applications and it is employed as a baseline method for comparison with other new methods. However, BR makes the basic assumption of complete independency between the assigned labels, which lowers the performance of the method for some applications.

This paper proposes a new ML classification method that uses BR at the core for its scalability. Our method defines a graphical-model-based extension to BR to reintroduce the dependency relationships between the labels lost by the main BR assumption. Our graphical model (Fig. 1) defines a new node for the output of each SC (S_i) classifier of BR. These nodes are observable binary random variables of our system because their values can be observed by running the corresponding SC classifier S_i. Our graphical model also includes a node that corresponds to the label assignment node, *LN*. Each node in the graphical model is dependent on the other nodes, which brings the dependency relationships into the ML classification system. The node *LN* is not directly observable, but it can be inferred using standard Bayesian inference rules. Our proposed system has many advantages. First, it can work on top

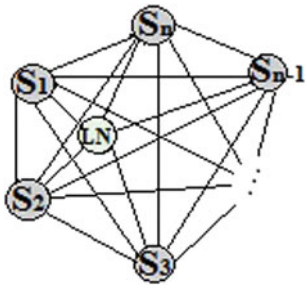


Fig. 1 The proposed graphical model. See text for details

of any BR system with different SC classifiers such as Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Decision Tree (DT), etc. Second, we can take advantage of the standard Bayesian network inference rules that can make decision even with partial observation data. Our method is scalable because it uses BR at its core. Finally, the implementation of the proposed system is very simple.

The rest of the paper is organized as follows. Section 1 introduces the background of ML and especially transform methods used. The Sect. 2 describes the proposed Graphical Model. The Sect. 3 lists the experiments , the datasets used in experiments, evaluation metrics, and experimental results with tables. The last section discusses the conclusion and points to future work.

2 Background

The ML classification problem can be defined with the help of the data matrix X of size $|D| \times |F|$, where $|D|$ is the number of examples of the training set and $|F|$ is the number of features for each sample. We also define a binary matrix Y of size $|D| \times |L|$, where $|L|$ is the size of the label set. We use the notation X_i and Y_i for the vectors to represent the ith row of the X and Y matrices, respectively. Similarly, we use the notation X_{ij} for the scalar to represent the jth feature of X_i and the notation $(Y_i)_j$ for the binary value to present the jth label of Y_i. The ML classification problem can be defined with the function f that produces a binary label vector Y_j for a given features vector X_j.

$$f : X_j \rightarrow Y_j. \tag{1}$$

For a given classification problem, if $|L|$ is 1, then the problem is called a SC classification problem. If $|L|$ is > 1 and only one element of the vector Y_j elements can be 1, then the problem is called MC classification. There are no restrictions on $|L|$ or the number of 1 s in Y_j for the ML classification, which shows that ML classification can be considered as the generalization of MC and SC.

In order to find the best function f that produces label values most similar to Y_i for a given X_i, the literature mostly uses transformation methods, which treats the ML classification as multiple SC and MC problems. The first transformation method, Pair-Wise (PW) [6, 11] transformation, treats each vector value as new class and performs MC classification. PW has problems with scalability because it needs to work with $2^{|L|}$ new labels, which can grow very fast. In addition, PW has some overfitting problems. Another transformation is the Label Combination(LC) [6, 11] transformation. LC takes sub-label vector combinations and assigns new labels for these combinations. LC shares the same scalability problems with the PW methods. The final transformation method is BR [1, 5], which defines a new SC classifier for each element of the vector Y_i. So there are $|L|$ SC classifiers. Each classifier is trained using one-versus-all strategy. In order to produce the final label vector Y_i, each classifier is run on vector X_i and the results of these classifications are catenated into Y_i. BR is very scalable because the number of classifiers increase linearly with the number of labels $|L|$. As a result, it is widely used. However, BR has made the assumption of complete label independence, which is not realistic. The next section introduces our graph-based method to address this problem.

3 The Graphical Model

We define a graph $G = \{V, E\}$ with vertex set $V = \{S_i, LN\}$ and edges between these vertices $E = \{E_{ij}, E_i\}$, where $i, j = 1 \cdots |L|$ (Fig. 1). S_i are the SC classifier graph nodes to represent the binary random variables that correspond to the BR classifier S for the label i. LN is the label node to represent the binary vector random variable that corresponds to the label vector Y_i. E_{ij} is the edge between the nodes S_i and S_j. Similarly, E_i is the edge between the nodes S_i and LN. Only the S nodes of graph G are observable. G is a fully connected graph which assumes that each label observation S_i is dependent on the other label observations and the node LN. The node LN is not directly observable. We use standard Bayesian network inference to obtain the binary vector LN values using

$$P(LN|S_1, S_2, \ldots, S_{|L|}) = \frac{P(LN, S_1, S_2, \ldots, S_{|L|})}{P(S_1, S_2, \ldots, S_{|L|)}} \tag{2}$$

The estimation of $P(LN, S_1, S_2, \ldots, S_{|L|})$ and $P(LN)$ values are done using standard Maximum Likelihood Estimation (MLE) on the Full Joint Probability Distribution Table (FJPDT) formed from the training data set D. This table is formed by adding rows that include the binary vector Y_i for a given sample D_i and the the results of SC_i classifications for the same sample. In other words, our table includes $|L| + 1$ columns and $|D|$ rows. The value of $P(S_1, S_2, \ldots, S_{|L|)})$ for a given values of S_1 to $S_{|L|}$ is estimated by MLE by

$$P(S_1, S_2, \ldots, S_{|L|}) = \frac{Count(S_1, S_2, \ldots, S_{|L|})}{|D|}, \tag{3}$$

which is the normalized count of the Y_j's of the dataset D. Similarly

$$P(LN, S_1, S_2, \ldots, S_{|L|}) = \frac{Count(Y_j, S1, \ldots, SL)}{|D|}, \tag{4}$$

which is the count of rows where Y_j and the results of SC classifiers occurs together. Note that our graphical model G potentially introduces $2^{|L|}$ new labels because the node *LN* can take binary vector values from a set of $2^{|L|}$ elements. However, the estimation of Eqs. 3 and 4 includes only operations that grow linearly with $|D|$. As a result, our proposed method is scalable and it addresses the label dependency problem by employing FJPDT. Although we used MLE for the estimation of the probability values, smoothing techniques can be employed to address any issues that involve count values of zeros. Note also that our method can estimate label information even with partial SC classification results using the standard Bayesian inference by enumeration.

4 Experiments

The literature defines standard evaluation metrics for the ML classification problems. [8] proposes Hamming-Loss metric that measures the normalized number of different elements between given two binary vectors such as Y_j and ground truth. Lower scores are better. The other metric is the accuracy [4], which measures the ratio of the intersection and the union between the estimated label set and the ground truth label set. If both sets are the same, then this value becomes 1, otherwise it gets smaller. The Exact-Matching [4] is the last metric used. It counts the normalized number of exact matches between the sets of estimated and the ground truth labels for a given sample. Exact-Matching can be at most 1 for the best estimation.

We used fivefold cross validation for all of our experiments. We defined an SVM based SC classifier for each label and used BR to transform each dataset problem listed in Table 1. Although we used SVM for the SC classification, any other SC

Table 1 Multi-label datasets

Names	|D|	|L|	|F|	LC	LD
Music	593	6	72num	1.87	0.31
Scene	2,407	6	294num	1.07	0.18
Yeast	2,417	14	103num	4.24	0.30
Emotions	593	6	72num	1.869	0.31
Birds	645	12	261num + 2b	1.014	0.053

The number of the samples |D|; the number of the Labels |L|; the number of the features |F|; b is for binary and num is abb. for numerical features, Label Cardinality (LC) and Label Density (LD)

Table 2 Hamming loss performance results

Names	SVM-BR	Our method
Music	0.187571	0.182203
Scene	0.119028	0.130556
Yeast	0.69834	0.745643
Emotions	0.471186	0.292655
Birds	0.23	0.13

Table 3 Accuracy performance results

Names	SVM-BR	Our Method	CC [6]	BR [6]	SMBR [6]	MBR [6]
Music	0.650±0.02	0.687±0.01	0.45±0.05	0.50±0.01	0.52±0.01	0.52±0.01
Scene	0.578±0.02	0.682±0.01	0.67±0.01	0.58±0.01	0.62±0.01	0.610,01
Yeast	0.48±0.01	0.51±0.01	0.50±0.01	0.51±0.01	0.50±0.01	–
Emotions	0.599±0.02	0.648±0.03	–	–	–	–
Birds	0.39±0.04	0.50±0.04	–	–	–	–

Table 4 Exact-matching performance results

Names	SVM-BR	Our method	CC [6]	BR [6]	SMBR [6]	MBR [6]
Music	0.29±0.02	0.3152±0.02	0.26±0.02	0.26±0.02	0.27±0.02	0.27±0.01
Scene	0.4262±0.01	0.5766±0.02	0.62±0.01	0.51±0.01	0.57±0.01	0.55±0.01
Yeast	0.1390±0.01	0.1452±0.01	0.20±0.02	0.15±0.01	0.18±0.01	0.15±0.01
Emotions	0.0390±0.01	0.0746±0.02	–	–	–	–
Birds	0.011±0.03	0.023±0.03	–	–	–	–

method can be employed with our method. Then we formed the FJPDT for each dataset and estimated the labels for the 20 % of the data for each cross validation step. For all experiments, we compared our performance with the base BR method that assumes full label independence.

The Hamming-Loss results for our experiments are listed at Table 2. Three of the five datasets produces better results for system with this metric. The Scene and Yeast datasets are worse with our method. Table 3 lists the accuracy numbers from our system along with the numbers from base BR method and other ML classification methods from [6]. As the accuracy numbers indicate, our method performs favorably for all datasets in terms of accuracy numbers. The final experiments show the exact-matching results at Table 4. Our system performs very well compared to the base BR method, and it is comparable with the state-of-the-art method of the [6]. Note that our proposed method is both simpler and more scalable than the method of [6].

5 Conclusion and Future Work

We presented a new ML classification method that uses BR for transforming the original problem into multiple SC classification problems. Our method is scalable like BR, but it also considers the label dependencies between the assigned labels. The label dependency is achieved by considering the SC classifiers as dependent observations of the samples. These observations are employed as nodes of a graphical model and the final label assignments are obtained using the standard Bayesian inference from the graphical model. Our method can be used with any type of SC classification method under BR such as SVM, KNN, or DT. In addition, our method can still work with a subset of SC classifier results. The experiments performed on standard multi-label data showed the effectiveness of the proposed method. For future work, we plan to extend the current system by using many SC classifiers for the same label and use similar Bayesian inference for a more robust label assignment.

References

1. E. Alvares-Cherman, J. Metz, M.C. Monard, Incorporating label dependency into the binary relevance framework for multi-label classification. Expert Syst. Appl. **39**(2), 1647–1655 (2012)
2. C. Bielza, G. Li, P. Larranaga, Multi-diimensional classification with bayesian networks. Int. J. Approx. Reason. **52**(66), 705–727 (2011)
3. A. Elisseeff, J. Weston, A kernel method for multi-labelled classification. in *Advances in Neural Information Processing Systems* 14, MIT Press, Cambidge, pp. 681–687 (2001)
4. S. Godbole, S. Sarawagi, Discriminative methods for multi-labeled classification. in Lecture Notes in Artificial Intelligence Subseries of Lecture Notes in Computer Science **3056**, 22–30 (2004)
5. O. Luaces, J. Dez, J. Barranquero, J.J. del Coz, A. Bahamonde, Binary relevance efficacy for multilabel classification. Prog. Artif. Intell. **1**(4), 303–313 (2012)
6. J. Read, Scalable multi-label classification Doctoral dissertation, University of Waikato (2010)
7. J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification. Mach. Learn. **39**(2), 135–168 (2000)
8. R.R. Schapire, Y. Singer, Improved boosting algorithms using confidence—rated predictions. Mach. Learn. **37**(3), 297–336 (1999)
9. X. Shen, M. Boutell, J. Luo, C. Brown, Multilabel machine learning and its application to semantic scene classification. in *Electronic Imaging* 2004 (pp. 188–199). International Society for Optics and Photonics (2003, December)
10. R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufis, D. Varga, The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. arXiv preprint cs/0609058 (2006)
11. G. Tsoumakas, I. Katakis, Multi-label classification: An overview. Int. J. Data Warehous. Min. (IJDWM) **3**(3), 1–13 (2007)
12. G. Thoumakas, I. Katakis, I. Vlahavas, Random k-labelsets for multi-label classification. IEEE Trans. Knowl. Discov. Data Eng. **23**(7), 1079–1089 (2010)
13. M.-L. Zhang, Z.-H. Zhou, ML-KNN: A lazy learning approach to multi-label learning. Pattern recognit. **40**(7), 2038–2048 (2007)

Ground Plane Detection Using an RGB-D Sensor

Doğan Kırcalı and F. Boray Tek

Abstract Ground plane detection is essential for successful navigation of vision based mobile robots. We introduce a very simple but robust ground plane detection method based on depth information obtained using an RGB-Depth sensor. We present two different variations of the method: the simplest one is robust in setups where the sensor pitch angle is fixed and has no roll, whereas the second one can handle changes in pitch and roll angles. Our comparisons show that our approach performs better than the vertical disparity approach. It produces accurate ground plane-obstacle segmentation for difficult scenes, which include many obstacles, different floor surfaces, stairs, and narrow corridors.

Keywords Ground plane detection · Kinect · Depth-map · RGB-D · Autonomous robot navigation · Obstacle detection · V-disparity

1 Introduction

Ground plane detection and obstacle detection are essential tasks to determine passable regions for autonomous navigation. To detect the ground plane in a scene the most common approach is to utilize depth information (i.e., depth map). Recent introduction of RGB-D sensors (Red-Green-Blue-Depth) allowed affordable and easy computation of depth maps. Microsoft Kinect— is the pioneer of such sensors— integrates an infrared (IR) projector, a RGB camera, a monochrome IR camera, a tilt motor and a microphone array to provide a 640×480 pixels depth map and RGB video stream at a rate of 30 fps.

Kinect uses an IR laser projector to cast a structured light pattern over the scene. Simultaneously, its monochrome CMOS IR camera acquires an image. The

D. Kırcalı (✉) · F.B. Tek
Robotics and Autonomous Vehicles Laboratory, Computer Engineering Department,
Işık University, 34980 Şile, İstanbul, Turkey
e-mail: dogan@isikun.edu.tr

F.B. Tek
e-mail: boray@isikun.edu.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_8

disparities between the expected and the observed patterns are used to estimate a depth value for each pixel. Kinect works well indoor. However, the depth reading is not reliable for regions that are far more than 4 m; at the boundaries of the objects because of the shadowing; reflective or IR absorbing surfaces; and at the places that are illuminated directly by sunlight which causes IR interference. Accuracy under different conditions was studied in [1–3].

Regardless of the method or the device that is used to obtain the depth map, several works approach to the ground plane detection problem based on the relationship between a pixel's position and it is disparity [4–9].

Li et al. show that the vertical position (y) of a pixel of the ground plane is linearly related to its disparity $D(y)$ such that one can seek a linear equation $D(y) = K1 + K2 * y$, where $K1$ and $K2$ are constants, which are determined by the sensor's intrinsic parameters, height, and tilt angle. However, ground plane can be directly estimated on the image coordinates using the plane equation based on disparity $D(x, y) = ax + by + c$ without determining mentioned parameters. A least squares estimation of the ground plane can be performed offline (i.e., by pre-calibration) if a ground plane only depth image of the scene is available [5]. Another common approach is to use RANSAC algorithm which allows fitting of the ground plane even the image includes other planes [4, 10, 11].

Some other approaches aim to segment the scene into relevant planes [11, 12]. The work of Holz et al. clusters surface normals to segment planes and reported to be accurate in close ranges [11].

In [7], histograms of the disparity image rows were used to model the ground plane. In the image formed of the row histograms (named as *V-disparity*), the ground plane appears as a diagonal line. This line, which is detected by Hough Transform, was used as the ground plane model.

In this paper, we present a novel and simple algorithm to detect the ground plane without the assumption that it is the largest region. Assuming a planar ground plane model which may probably cause problems if the floor has significant inclination or declination [6, 7], we use the fact that if a pixel is from the ground plane, its depth value must be on a rationally increasing curve placed on its vertical position. Although the degree of this curve is not known, it can be estimated by an exponential curve fit to use it as the ground plane model. Later, the pixels which are consistent with the model are detected as ground plane whereas the others are marked as obstacles. While this is our base model which can be used for a fixed viewing angle scenario, we provide an extension of it for dynamic environments where sensor viewing angle changes from frame to frame. Moreover, we note the relation of our approach to V-disparity approach [7], which rely on the linear increase of disparity and fitting of a line to model the ground plane, and compare our method by tests on the same data.

2 Method

2.1 Detection for Fixed Pitch

In a common scenario, the sensor views the ground plane with an angle (i.e., pitch angle), in which we can assume that the sensor is fixed and its roll angle is zero (Fig. 1b). The sensor's pitch angle (Fig. 1a) causes allocation of more pixels for the closer scene than the farther. So that linear distance from the sensor is projected on the depth map as a rational function. This is demonstrated in (Fig. 1c). Any column of the depth image shows that the depth value increases not linearly but rationally from bottom to top (i.e., right to left in Fig. 1d). Furthermore, a "ground plane only" depth image must have all columns equal to each other, which is estimable by a curve fit of sum of two exponential functions in the following form:

$$f(x) = ae^{bx} + ce^{dx} \tag{1}$$

where $f(x)$ is the pixel's depth value and x is the its vertical location (i.e., row index) in the image. The coefficients (a, b, c, d) depend on the intrinsic parameters, pitch angle, and the height of the sensor.

A least squares fitting estimation of these coefficients make it possible to reconstruct a curve, which is named as the *reference ground plane curve* (C_R). In order to detect ground plane pixels in a new depth frame, its columns(C_U) are compared to C_R; any value under C_R represents an object (or a protrusion), whereas values above

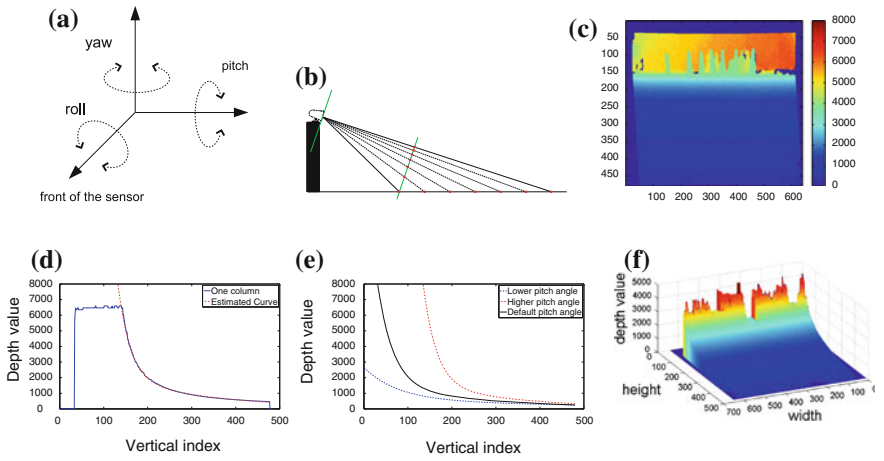


Fig. 1 **a** Roll & pitch axis, **b** sensor view pitch causes linearly spaced points to mapped as an exponential increasing function. **c** An example depth map image, **d** one column (y = 517) of the depth map and its fitted *curve* representing the ground plane, **e** ground plane curves for different pitch angles, **f** depth map in three dimensions showing the drop-offs caused by the objects

the reference curve represent drop-offs, holes (e.g., intrusions, downstairs, edge of a table). Hence, we compare the absolute difference against a pre-defined threshold value T; mark the pixels as ground plane if difference is less than T. Here, the depth values that are equal to zero were ignored as they indicate sensor reading errors. The related experiments are in Sect. 3.

2.2 Detection for Changing Pitch and Roll

The fixed pitch angle scheme explained above is quite robust. However, it is not suitable for the scenarios where the pitch and roll angles of the sensor changes. Such as of the mobile robots that cause movements on the sensors' platform. These can be compensated by using an additional gyroscopic stabilization [13]. However, here we propose a computational solution in which a new reference ground curve is estimated for each new input frame.

A higher pitch angle (sensor almost parallel to the ground) will increase the slope of the ground plane curve. Whereas a non-zero roll angle (horizontal angular change) of the sensor forms different ground plane curves along columns of the depth map (Fig. 1e). Such that at one end the depth map exhibits curves of higher pitch angles, while toward the other end, it has curves of lower pitch angles, which complicate the use of a single reference curve for that frame.

To overcome the roll angle effects our approach aims to rotate the depth map to make it orthogonal to the ground plane. If the sensor is orthogonal to the ground plane it is expected to produce equal or very similar depth values along every horizontal line (i.e., rows), which can be captured by a histogram of the row values such that a higher histogram peak value indicates more similar values along a row. Let h_r shows the histogram of the rth row of a depth image (D) of R rows, and let us denote the rotation of depth image with D_θ.

$$argmax_\theta\left(\sum_{r=1}^{R} argmax_i(h_r(i, D_\theta)\right) \tag{2}$$

Thus for each angle value θ in a predefined set, the depth map is rotated with an angle θ and the histogram h_r is computed for every row r. Then, the angle θ that maximizes the sum of the histogram peak values is estimated as the best angle to rotate the depth map prior to the ground plane curve estimation. This removes the roll angle effect.

The changes of pitch angle create different projection and different curves along the image columns (Fig. 1e). However, in a scene that consists of both the ground plane and objects, the maximum value along a particular row of the depth map must be due the ground plane, unless an object is covering the whole row (as in Fig. 1f). This is because the objects are closer to the sensor than the ground plane surface that they occlude. Therefore, if the maximum value across each row (r) of the depth map

(D) is taken, which we name as the *depth envelope* (E), it can be used to estimate the *reference ground plane curve* (C_R) for this particular scene and frame.

$$E(r) = max_i(D(c_i, r)) \tag{3}$$

The estimation is again performed by fitting the aforementioned exponential curve (1). Prior to the curve fitting we perform median filtering to smooth the depth envelope. Moreover, depth values must increase exponentially from bottom of the scene to the top. However, when the scene ends with a wall or group of obstacles this is reflected as a plateau in the depth envelope. Hence, the envelope (E) is scanned from right to left and the values after the highest peak are excluded from fitting as they cannot be a part of the ground plane. After the curve is estimated pixels of the frame are classified, as described in Sect. 2.1.

Two conditions affect the ground plane curve fit adversely. First, when one or more objects cover an entire row, this will produce a plateau in the profile of the depth map. However, if the rows of the "entire row covering object or group" do not form the highest plateau in the image, ground plane curve continues afterwards and the object will not affect the curve estimation. Second, drop-offs of the scene cause sudden increases (hills) on the depth envelope because they exhibit depth values higher than the ground plane's: If a hill is found on the depth envelope, the estimated curve will be produced by a higher fitting error.

3 Experiments

We tested our algorithm on four different datasets comprised of several 640×480 frames. Dataset-1 and dataset-2 were composed of 300 frames captured on a robot platform which moves on the floor among several obstacles. Dataset-3 was created with the same platform; however, the pitch and roll angles change excessively. Dataset-4 included 12 individual frames acquired from difficult scenes such as narrow corridors, wall only scenes etc. Dataset-1 and dataset-2 were manually labeled to provide the ground truth and were used in plotting ROCs (Receiver Operating Curve), whereas the other two were visually examined.

We compared three different versions for our approach: fixed pitch (A1), pitch compensated (A2), pitch and roll compensated (A3). There is only one free parameter for A1 and A2 that is the threshold T, which is estimated by ROC analysis, whereas the third roll compensation algorithm requires a pre-defined angle set for the search for best rotation angle: $\{-30°, -28°, \ldots, +30°\}$. Least squares fit was performed by Matlab curve fitting function with default parameters.

Moreover, we compared the results with V-disp method [7], which is originally developed for stereo depth calculation where the disparity is available before depth. To implement V-disp method, we calculated disparity from the Kinect depth map (i.e., $1/D$); calculated the row histograms to form V-disp image; and then run Hough

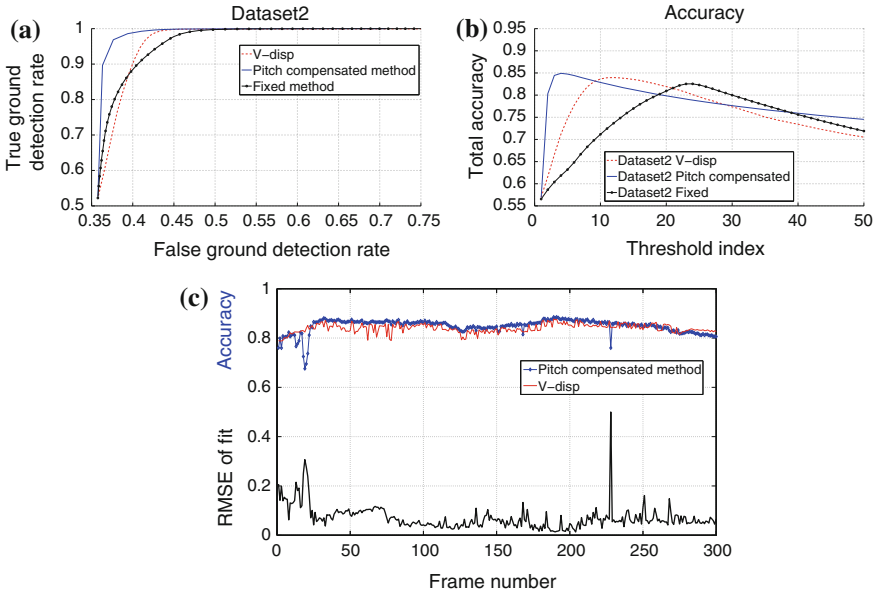


Fig. 2 **a** ROC curves comparing V-disp and our fixed and pitch compensated algorithms (A1-A2), **b** average accuracy over 300 frames versus thresholds, **c** accuracy and *curve* fit error of A2 for individual frames

transform to estimate the ground plane line. We put a constraint on the Hough line search in $[-60°, -30°]$ range.

Since A3 and A2 algorithms are same, except for the roll compensation, we examine and compare results of A2 to A1 and V-disp; however, we compare A3 results only against A2 to show the effect of the roll compensation.

Figure 2a, b show ROC curves and overall accuracies plotted for our fixed and pitch compensated algorithms (A1 and A2) and V-disp method on dataset-2. It can be seen that our pitch compensated algorithm is superior to V-disp which is better than our fixed algorithm.

When we select the best accuracy point thresholds and run our algorithms on dataset-2, we obtained accuracy versus frames (Fig. 2c). In addition, we recorded the curve fitting error for the pitch compensated algorithm (A2). Both methods were quite stable with the exception of some high curve fitting error frames for A2. Those frames can be automatically rejected to improve accuracy.

Some example inputs and outputs of our algorithm A2 is shown in Fig. 3. The examples include a cluttered scene (Fig. 3a–c), stairs (Fig. 3d–f), one of the frames from dataset-3, where the sensor is rolled almost 20° degrees (Fig. 3g). Figure 3h, i shows the respective outputs of A2 and A3. It can be seen that the roll compensation provides a significant advantage.

Finally, Fig. 3j, k shows output pairs (overlaid on RGB) for A2 and V-disp. Both methods detect the ground planes in the scenes where ground plane is not the largest

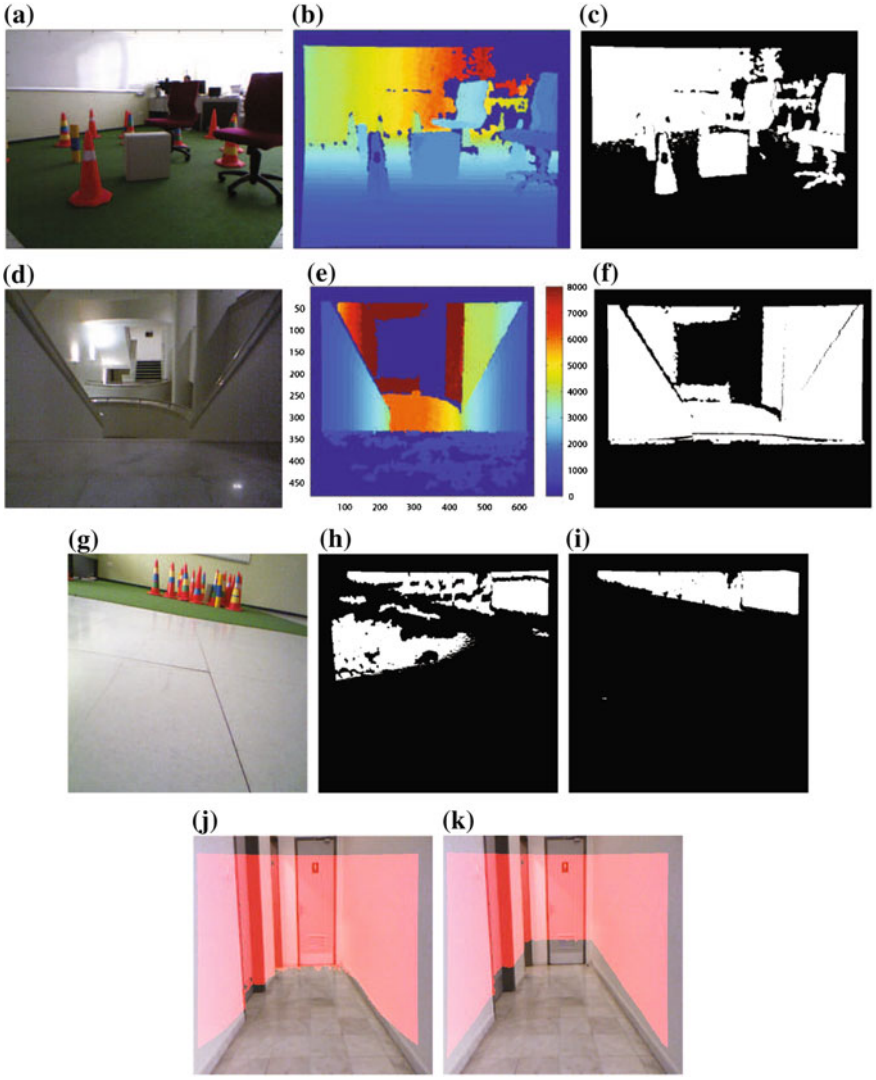


Fig. 3 Experimental results from different scenes. RGB, depth-map and pitch compensated method output (*white* pixels represent objects whereas black pixels represent ground plane): **a**, **b**, **c** lab environment with many objects and reflections; **d**, **e**, **f** stairs **g**, **h**, **i** respective outputs of pitch compensated (A-2) and pitch&roll compensated method on an image where sensor was positioned with a roll angle (A-3). **j**, **k** Comparison of pitch compensated (*left*) and V-disp method (*right*) in narrow corridor

nor the dominant plane. Note that A2 is better than V-disp, though the thresholds used by both methods were determined for the highest respective overall accuracy for dataset-1, -2.

If the frames are buffered beforehand, our algorithm A2 processed 83 fps on a Pentium i5 processor using Matlab 2011a. Datasets and more results can be found in our web site.

4 Conclusion

We have presented a novel, and robust ground plane detection algorithm which uses depth information obtained from an RGB-D sensor. Our approach includes two different methods, where the first one is simple but quite robust for fixed pitch and no-roll angle scenarios, whereas the second one is more suitable for dynamic environments. Both algorithms are based on an exponential curve fit to model the ground plane which exhibits rationally increasing depth values. We compared our method to the popular V-disp [7] method which is based on detection of a ground plane model line by Hough transform which relied on linear increasing disparity values. We have shown that the proposed method is better than V-disp and produces acceptable and useful ground plane-obstacle segmentation for many difficult scenes, which included many obstacles, different surfaces, stairs, and narrow corridors.

Our method produce errors especially when the curve fitting is not successful. Our future work will focus on these situations that are easy to detect by checking the RMS error of the fit, which has been shown to be highly correlated with the accuracy of segmentation.

Acknowledgments This study was supported by FMV Işık University Internal Research Funding Grants project BAP-10B302.

References

1. J. Stowers, M. Hayes, A. Bainbridge-Smith, Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor. in *IEEE International Conference on Mechatronics* (ICM), pp. 358–362 (2011)
2. C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, J. Meunier, Fall detection from depth map video sequences. in *International Conference on Smart Homes and Health Telematics*, pp. 121–128 (2011)
3. K. Khoshelham, S.O. Elberink, Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors **12**(2), 1437–1454 (2012)
4. F. Li, J.M. Brady, I. Reid, H. Hu, Parallel image processing for object tracking using disparity information. in *In 2nd Asian Conf. on Computer Vision ACCV*, pp. 762–766 (1995)
5. S. Se, M. Brady, Ground plane estimation, error analysis and applications. Robot. Auton. Syst. **39**(2), 59–71 (2002)
6. Q. Yu, H. Araújo, H. Wang, A stereovision method for obstacle detection and tracking in non-flat urban environments. Auton. Robots **19**(2), 141–157 (2005)
7. R. Labayrade, D. Aubert, J.P Tarel, Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. in *IEEE Intelligent Vehicle Symposium*, vol. 2, pp. 646–651 (2002)

8. C.J. Taylor, A. Cowley, Parsing indoor scenes using rgb-d imagery. in *Robotics: Science and Systems* July 2012
9. K. Gong, R. Green, Ground-plane detection using stereo depth values for wheelchair guidance. in *Proceedings 7th Image and Vision Computing New Zealand*, pp. 97–101 (2009)
10. C. Zheng, R. Green, Feature recognition and obstacle detection for drive assistance in indoor environments. in *Proceedings 11th Image and Vision Computing New Zealand* (2011)
11. D. Holz, S. Holzer, R. Bogdan Rusu, S. Behnke, Real-Time Plane Segmentation using RGB-D Cameras. in *Proceedings of the 15th RoboCup International Symposium*, vol. 7416, pp. 307–317, Istanbul, Turkey, July 2011
12. C. Erdogan, M. Paluri, F. Dellaert, Planar segmentation of rgbd images using fast linear fitting and markov chain monte carlo. in *The 12' Conference on Computer and Robot Vision*, pp. 32–39 (2012)
13. L. Wang, R. Vanderhout, T. Shi, Computer vision detection of negative obstacles with the microsoft kinect. *Uni. British Columbia. ENPH 459 Reports* (2012)

An Iterated Local Search Platform for Transportation Logistics

Takwa Tlili and Saoussen Krichen

Abstract Recent technological advances in optimization and transportation have enabled the development of efficient tools that support the decision-making process of logistic managers. The aim of this paper is to present a decision support system (DSS) that integrates an Iterated Local Search (ILS) for solving the Vehicle Routing Problem (VRP). Its clear design allows an efficient exploration of the solution by the decision-maker. The computational experiments show that the ILS is very competitive in comparison to state-of-the-art algorithms.

Keywords Iterated local search · Decision support system · Vehicle routing problem

1 Introduction

The efficient transportation management holds significant value due to its high impact on customer satisfaction by reducing delivery costs. Bearing in mind that logistic problems are generally modeled as VRPs, such a formulation contributes directly to minimize costs of all such processes [2].

Routing problems have drawn the attention of many researchers due to the increasing concerns on economic and environmental problems. The VRP seeks the tours of minimum cost such that the vehicles start their trips from a central depot and turn back after serving all vertices in the network.

Numerous variations of the classical VRP have been introduced by modifying or adding additional attributes on solutions construction. The asymmetric VRP (AVRP) is concerned with finding a solution in which the distance between each pair of nodes in the two directions is not the same [3]. The Open VRP (OVRP) is the variant where the vehicles are not required to come back to the depot after completing their services [8, 11, 14]. The Multi-Depot VRP (MDVRP) is a generalization of the problem where more than one depot may be considered [21]. More recently, other VRP variants have

T. Tlili (✉) · S. Krichen
LARODEC, Institut Supérieur de Gestion Tunis, Université de Tunis, Tunis, Tunisia
e-mail: takwa.tlili@gmail.com

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_9

been evoked such as Split Delivery VRP [4], rollon-rolloff VRP [22] and Multi-Trip VRP [6].

To cup with the routing variants, a variety of exact approaches have been devoted but optimal solutions are only limited to small-scaled instances. Given the fact that the VRP is NP-hard [11], several approximate algorithms have been proposed for finding optimal routes within acceptable computational time. Metaheuristics, including Simulated Annealing [9], Genetic Algorithm [18, 20], Particle swarm optimization [1, 15, 17] have shown advantages in handling the VRPs. A recent comprehensive overview on metaheuristics for solving the VRP can be found in the work of Szeto et al. [19].

Nowadays, many industrial firms emphasize the need to use a decision support tool for solving VRPs. For illustration, E-Mendoza et al. [16] proposed a DSS that integrates SAP/R3 and ArcGIS to handle the distance-constrained VRP. Manzini [13] developed a DSS based on the top-down and multistep approach to handle the management of logistic networks. The dynamic VRP variant has been solved by Dahl and Derigs [7] using a web-based DSS that integrates decentralized databases and an online heuristic.

The main contributions of this paper are (i) to solve the VRP using a modified ILS metaheuristic evaluated on several VRP benchmarks. (ii) to design a DSS that computationally outperforms other state-of-the-art approaches. The remaining of the paper is organized as follows: Sect. 2 presents a detailed explanation of the proposed ILS algorithm. In Sect. 3 a description of the DSS architecture is provided. In Sect. 4 computational experiments highlight the effectiveness of the approach.

2 Architecture of the Decision Support System

Decision Support Systems are computer-based information systems that aid different business or organizational processes involving decision-making. The conceptual design of the proposed DSS for integrated optimization routines is shown in Fig. 1, that encompasses four main steps: orders data extraction, geographical data extraction, optimization analysis and display geographical output.

① **Orders Data**

The DSS starts by the extraction of orders data for each customer to serve. The weights of cargos to deliver is known in order to fulfill vehicles capacity constraints.

② **Routing Data**

The DSS extracts the routing data from the geographical database to construct the adjacency matrix. The geographical data and transport information are the inputs for the resolution step.

The two steps, Orders Data and Routing Data are achieved through the interface presented in Fig. 2 that allows the users to manager all inputs.

③ **Resolution process**

Once all required information are sent to the optimization engine, the resolution process is initiated. The screenshot reported in Fig. 3 shows that the decision maker can select the solution approach that can be either CPLEX or ILS metaheuristic.

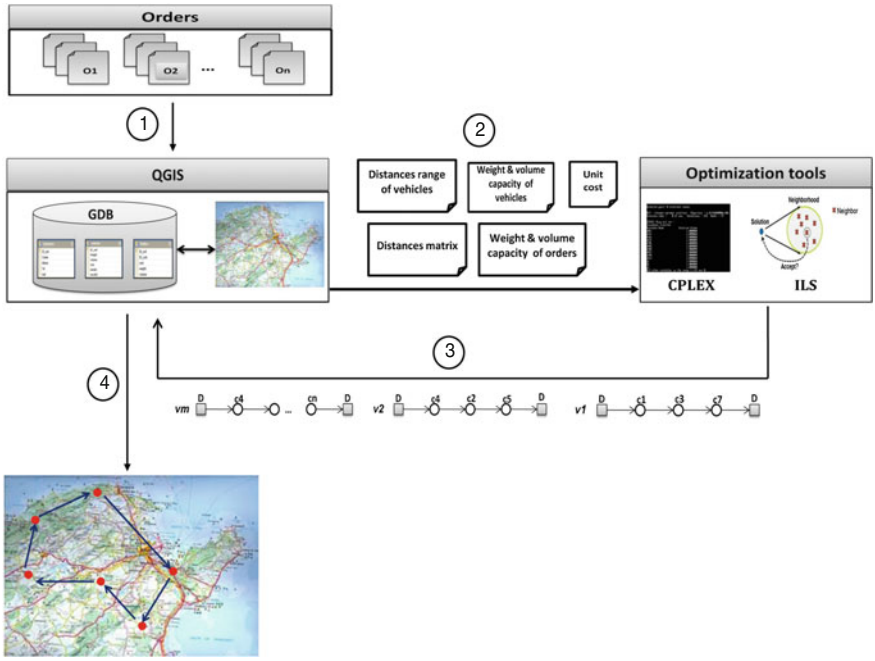


Fig. 1 Decision support system design

④ **Display geographical output**

The solution obtained from the optimization engine is transformed to a geographical solution presented as routes on maps as shown in Fig. 4.

3 Iterated Local Search Approach

Iterated local search (ILS) approach, introduced by Lourenço et al. [12], is applied to numerous combinatorial optimization problems successfully. Its performance is very competitive compared to other state-of-the art metaheuristics, such as particle swarm optimization, simulated annealing, and genetic algorithm.

The ILS is about combining phases of local search around the current solution and perturbations to diversify the search and avoid—local optima. The algorithm starts its search from an initial solution S_0 computed by a greedy heuristic. This solution is then improved to obtain a first local optimum S using the local search procedure. Iteratively, a perturbation is applied to the current solution with the hope to escape from its attraction basin. The execution of the algorithm is stopped when a maximum number of iterations is reached.

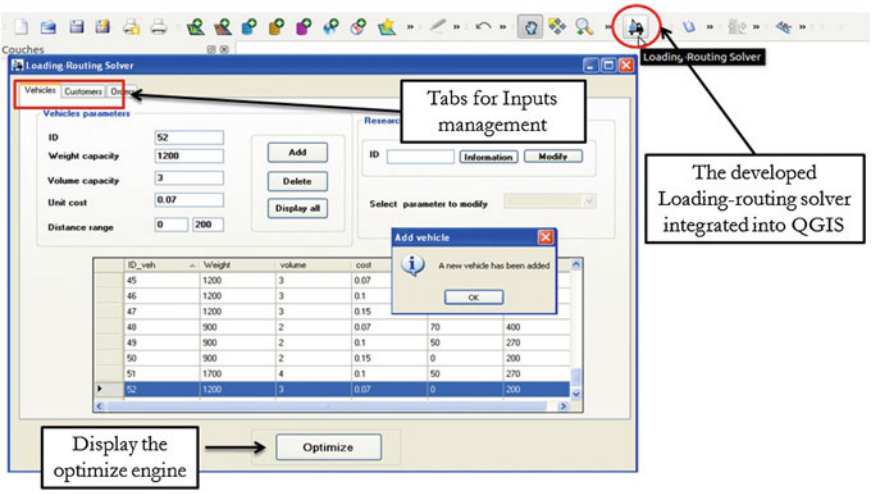


Fig. 2 Interface of the SDSS integrated in QGIS

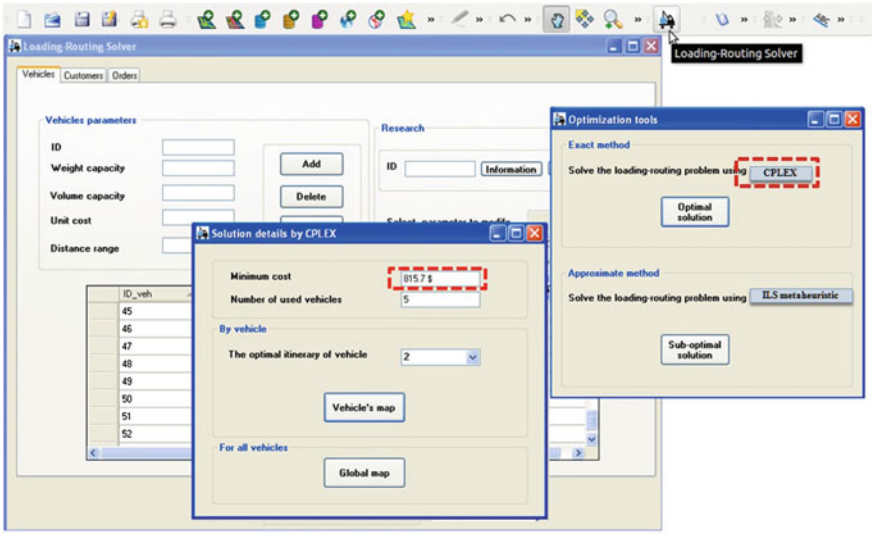


Fig. 3 Selecting either exact or approximate method

The pseudocode of the ILS algorithm [10] that we implemented is sketched in Algorithm 1.

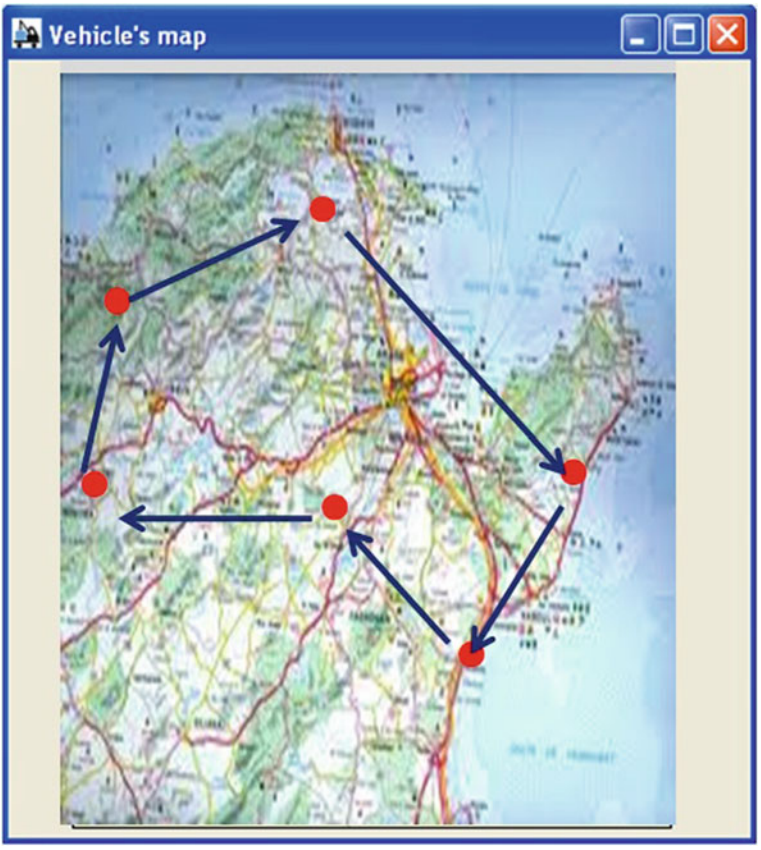


Fig. 4 Geographical solution mapped

Algorithm 1 ILS pseudocode

```
1: S_0 ← Initial_solution();
2: S ← Local_search(S_0);
3: repeat
4:    S' ← Perturbation (S);
5:    S'' ← Local_search(S');
6:    if f(S'') < f(S) then
7:       S ← S'';
8:    end if
9: until termination condition met
```

Table 1 Computational results on benchmark problems

Instance	Best	Travel cost		CPU time (s)		Instance	Best	Travel cost		CPU time (s)	
		GA	ILS	GA	ILS			GA	ILS	GA	ILS
A-n32-k5	784	957	1138	226	7	**A-n45-k6**	944	1452	**1277**	326	**5.59**
A-n33-k5	661	781	816	233	124.48	**A-n45-k7**	1146	1397	1573	333	22.2
A-n33-k6	742	798	877	266	92.98	**B-n31-k5**	672	736	**699**	241	**37.28**
A-n34-k5	778	923	925	264	3.0	**B-n34-k5**	788	865	879	264	81.65
A-n36-k5	799	1019	**989**	275	**17.09**	**B-n35-k5**	955	1263	**1066**	274	**3.17**
A-n37-k5	669	959	**934**	287	**39.57**	**B-n38-k6**	805	925	**920**	284	**40.98**
A-n37-k6	949	1115	1280	287	12.6	**B-n39-k5**	549	744	769	296	137.16
A-n38-k5	730	974	**972**	280	**130.52**	**B-n41-k6**	829	1105	**892**	311	**5.11**
A-n39-k5	822	1095	**1090**	286	**253.11**	**B-n43-k6**	742	1047	**885**	315	**4.38**
A-n39-k6	831	1257	**1128**	302	**97.36**	**B-n44-k7**	909	1126	1210	332	4.64

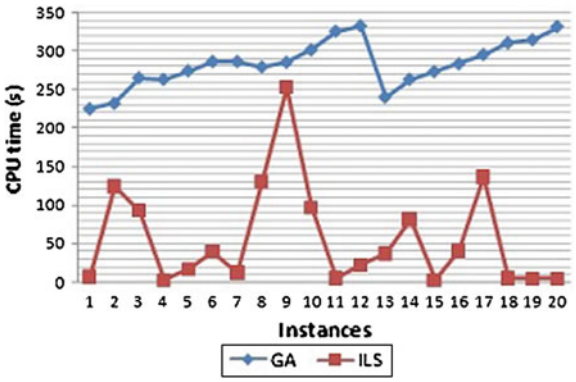


Fig. 5 CPU time of GA and ILS

4 Simulation

We analyze in what follows the results obtained on a set of experiments conducted to evaluate the performance of the developed algorithm. The ILS is coded in the Java programming language and experimentally evaluated on a laptop equipped with Core i5 processor and 2.5 GHz and 8 GB of RAM. The ILS is applied to some benchmark problems available in http://www.branchandcut.org/VRP/data. We select 20 problems from [5] benchmark, in which the number of customers is between 31 and 45.

Table 1 shows the experimental results on the instances and compares the performances of the ILS with the Genetic algorithm (GA) of Tsan and Gen [20]. In Table 1, we report the travel costs as well as the CPUs of ILS and GA. It can be seen that the ILS algorithm, in half of the instances outperformed the GA for both travel cost and CPU time. In terms of runtime, our method is quite faster than the GA for the whole set of problems as shown in Fig. 5.

5 Conclusion

In this paper, a prototype routing decision support system has been proposed in order to provide assistance to operating managers in transportation logistics. The architecture of the DSS consists of two key components: the geographical system and the optimization engine that applies a specifically designed ILS for solving the VRP. Comprehensive computational experiments and comparisons to an existing genetic algorithm showed that the ILS algorithm performs impressively, in terms of both solution quality and computational efficiency.

References

1. T.J. Ai, V. Kachitvichyanukul, Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. Comput. Ind. Eng. **56**, 380–387 (2009)
2. C. Alabas-Uslu, B. Dengiz, A self-adaptive local search algorithm for the classical vehicle routing problem. Expert Syst. Appl. **38**, 8990–8998 (2011)
3. S. Almoustafa, S. Hanafi, N. Mladenovic, New exact method for large asymmetric distance-constrained vehicle routing problem. Eur. J. Oper. Res. **226**, 386–394 (2013)
4. C. Archetti, N. Bianchessi, M.G. Speranza, Branch-and-cut algorithms for the split delivery vehicle routing problem. Eur. J. Oper. Res. **238**, 685–698 (2014)
5. P. Augerat, Approche Polyhédrale du Problème de Tournées de Véhicules Ph.D. Dissertation, Institut Polytechnique de Grenoble (1995)
6. D. Cattaruzza, N. Absi, D. Feillet, T. Vidal, A memetic algorithm for the multi trip vehicle routing problem. Eur. J. Oper. Res. **236**, 833–848 (2014)
7. S. Dahl, U. Derigs, Cooperative planning in express carrier networks? an empirical study on the effectiveness of a real-time decision support system. Decis. Support Syst. **51**, 620–626 (2011)
8. C. Erbao, L. Mingyong, Y. Hongming, Open vehicle routing problem with demand uncertainty and its robust strategies. Expert Syst. Appl. **41**, 3569–3575 (2014)
9. N. Ghaffari-Nasab, S.G. Ahari, M. Ghazanfari, A hybrid simulated annealing based heuristic for solving the location-routing problem with fuzzy demands. Sci. Iranica **20**, 919–930 (2013)
10. A. Grosso, A. Jamali, M. Locatelli, Finding maximin latin hypercube designs by iterated local search heuristics. Eur. J. Oper. Res. **197**, 541–547 (2009)
11. J. Lenstra, A. Kan, Complexity of vehicle routing and scheduling problems. Networks **11**, 221–227 (1981)
12. H. Lourenço, O. Martin, T. Stutzle, Iterated local search, *Handbook of Metaheuristics of International Series in Operations Research & Management Science* (Kluwer Academic, Dordrecht, 2003), pp. 321–353
13. R. Manzini, A top-down approach and a decision support system for the design and management of logistic networks. Transp. Res. Part E **48**, 1185–1204 (2012)
14. Y. Marinakis, M. Marinaki, A bumble bees mating optimization algorithm for the open vehicle routing problem. Swarm Evol. Comput. **15**, 80–94 (2014)
15. Y. Marinakis, M. Marinaki, G. Dounias, A hybrid particle swarm optimization algorithm for the vehicle routing problem. Eng. Appl. Artif. Intell. **23**, 463–472 (2010)
16. J.E. Mendoza, A.L. Medaglia, N. Velasco, An evolutionary-based decision support system for vehicle routing: the case of a public utility. Decis. Support Syst. **46**, 730–742 (2009)
17. S. MirHassani, N. Abolghasemi, A particle swarm optimization algorithm for open vehicle routing problem. Expert Syst. Appl. **38**, 11547–11551 (2011)
18. P.C. Pop, O. Matei, C.P. Sitar, An improved hybrid algorithm for solving the generalized vehicle routing problem. Neurocomputing **109**, 76–83 (2013)

19. W. Szeto, Y. Wu, S.C. Ho, An artificial bee colony algorithm for the capacitated vehicle routing problem. Eur. J. Oper. Res. **215**, 126–135 (2011)
20. A.S. Tasan, M. Gen, A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. Comput. Ind. Eng. **62**, 755–761 (2012)
21. W. Tu, Z. Fang, Q. Li, S.-L. Shaw, B. Chen, A bi-level voronoi diagram-based metaheuristic for a large-scale multi-depot vehicle routing problem. Transp. Res. Part E **61**, 84–97 (2014)
22. J. Wy, B.-I. Kim, A hybrid metaheuristic approach for the rollon-rolloff vehicle routing problem. Comput. Oper. Res. **40**, 1947–1952 (2013)

On Fuzzy Extensions to Energy Ontologies for Text Processing Applications

Dilek Küçük, Doğan Küçük and Adnan Yazıcı

Abstract Ubiquitous application areas of domain ontologies include text processing applications like categorizing related documents of the domain, extraction of information from these documents, and semantic search. In this paper, we focus on the utilization of two energy ontologies, one for electrical power quality and the second for wind energy, within such applications. For this purpose, we present fuzzy extensions to these domain ontologies as fuzziness is an essential feature of the ultimate forms of the ontologies to enhance such text analysis applications in the energy domain. A text categorization system for scholarly articles based on the extended power quality ontology is also presented for illustrative purposes together with its performance evaluation results.

1 Introduction

A domain ontology is defined as the set of concepts with their properties, their interrelationships, and constraints on these concepts that represent the semantics of the domain under consideration [1]. Such ontologies have been proposed for various domains including bioinformatics, chemical process engineering, electrical power quality, and wind energy, as reported in [2–5], respectively. These domain ontologies have significant application areas including those related to Semantic Web as surveyed in [6] and other text analysis applications. There are also fuzzy ontology proposals as described in [7] used for medical information retrieval, in [8] for news summarization, and in [9] for acquiring and sharing scientific information on the Web.

D. Küçük (✉)
Electrical Power Technologies Group, TÜBİTAK Energy Institute, Ankara, Turkey
e-mail: dilek.kucuk@tubitak.gov.tr

D. Küçük
Department of Computer Engineering, Technology Faculty, Gazi University, Ankara, Turkey
e-mail: dogan.kucuk@gazi.edu.tr

A. Yazıcı
Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
e-mail: yazici@ceng.metu.edu.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_10

The need for the representation of semantic information in energy-related domains has considerably increased particularly due to (i) the recent emergence and widespread implementation of smart grid applications[1] and (ii) the realization of the advantages of the renewable energy plants (like wind power and solar plants) over conventional fossil-fueled plants and hence the accelerating increase of the share of the renewable energy sources in energy generation all over the world.

In this study, we present fuzzy extensions to two ontologies from the energy domain: the first one is the ontology for electrical power quality [4] and the second one is the ontology for wind energy [5]. We describe our procedure to create fuzzy versions of these domain ontologies in order to make better use of them within text processing applications. The final forms of the extended ontologies are made publicly available for research purposes.

The rest of the paper is organized as follows: In Sect. 2, after brief descriptions of these energy-related ontologies, the main motivation behind the fuzzy extensions to these ontologies is presented. Section 3 describes the actual extension procedure on the ontologies making use of Wikipedia disambiguation pages. A text categorization application which illustrates the employment of the extended power quality ontology is described in Sect. 4 with its performance evaluation results and finally Sect. 5 concludes the paper.

2 Motivation

The domain ontology for electrical power quality (PQONT), as presented in [4], covers the concepts regarding mainly the power quality parameters like frequency, flicker, voltage/current harmonics, and events. PQONT is manually engineered with the experts of the domain. The taxonomy of the concepts within PQONT corresponding to these parameters is presented in Fig. 1, as excerpted from [4].

Each of these classes in PQONT has a *label* and a *synonymSet* attribute in which the natural language expression (in English) for the class and the synonyms of the expression are modeled, respectively. These two attributes are employed in a natural language interface for querying power quality data from a database where the queries are spelled in natural language. The aforementioned attributes of the PQONT concepts are used to determine the queried power quality parameter(s) within the query expression posed through the interface.

In order to extend PQONT to languages other than English, similar two attributes for these new languages can be added to PQONT. For instance, to hold the values for these two attributes in Turkish, the attributes of *translationInTurkish* and *synonymSetInTurkish* are included in the PQONT ontology [4].

Wind energy ontology (WONT) is created though a semi-automated procedure: first, related Wikipedia articles are automatically analyzed to determine the most

[1] Smart grid is usually defined as an enhanced electricity grid in which the components involved like the substations, centers, and the communication framework are more automated, reliable, and efficient since information and communication technology is employed [10, 11].

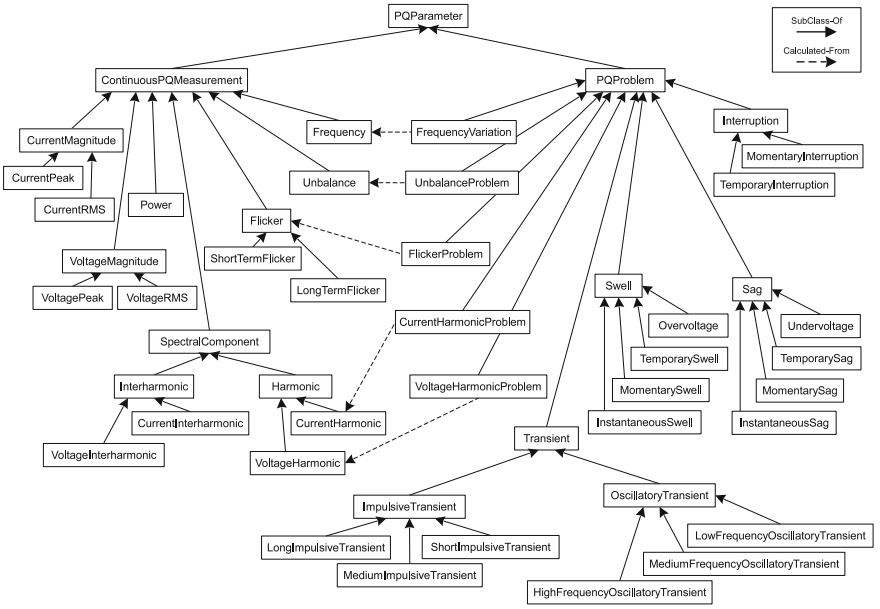


Fig. 1 The taxonomy of the *PQParameter* concept within PQONT

frequent n-grams used within the articles, then these n-grams are reviewed and the ultimate ontology is built by using the relevant n-grams and other semantic information sources [5]. The taxonomy of the concepts of the ontology, together with their interrelationships, is given in Fig. 2. Similar to PQONT, WONT also has the

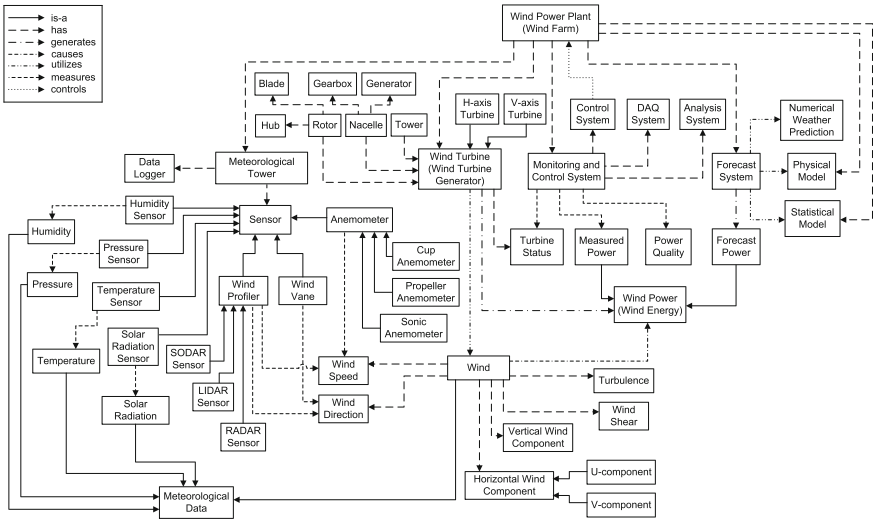


Fig. 2 The taxonomy of the concepts within WONT

label and *synonymSet* attributes to facilitate the utilization of the ontology within text analysis applications.

Both PQONT and WONT are significant semantic information sources for the related energy domains and such sources are quite rare within the energy domain. As pointed out in Sect. 1, they can be utilized in Semantic Web applications, knowledge-based systems, and text analysis applications.

In this paper, we consider the possible contribution of ontologies to diverse text analysis applications[2] through the related attributes of *label* and *synonymSet*. These attributes within PQONT have been seamlessly used in a natural language interface for a power quality database [4]. This interface is a domain-specific application where the possible homonymy of the words/phrases with other common names do not lead to problems. Yet, in several open-domain applications like text categorization or information extraction systems this homonymy may cause problems on diverse text types, as outlined below:

- The *Power* concept of PQONT has *power* as the value of its *label* attribute. It is defined as "the amount of work done by an electric current per unit time or the rate at which electrical energy is transferred". But the word (*power*) has other uses (i.e., is homonymous to other common names) corresponding to "physical force or strength" or "political influence". Hence, the appearance "power" in a document in itself cannot be considered as a sign of the document's belonging to the topic of electrical power quality with 100 % confidence, yet the document can be considered relevant with a confidence less than 100 %.
- The *Pressure* concept of WONT has the *label* value of "pressure" which has other meanings like "distress" or "urgency". Accordingly, it can only be considered a relevant keyword for the wind energy domain with a confidence less than 100 % instead of a confidence of strict 100 %.
- The above observations are also applicable to the values in the list-valued *synonymSet* attributes of the concepts within the two ontologies in addition to the values of their *label* attributes.

Hence, to make use of these two energy-related ontologies in open-domain text processing applications, we have extended the ontologies by adding as new attributes the membership values corresponding to the *label* and *synonymSet* attributes of the concepts. The details of this extension procedure are presented in the following section.

3 Fuzzy Extensions to the Power Quality and Wind Energy Ontologies

In order to extend the PQONT and WONT ontologies, we have added two new attributes to the concepts of these ontologies. The first new attribute, *membership ValueLabel*, is a numerical attribute of floating point type which can take a value from

[2] Depending on the interpretation, text analysis applications can well be considered as part of Semantic Web applications or knowledge-based systems.

the interval [0–1] and holds the relevance of the value of the *label* attribute of the concept to the domain of the ontology under consideration. The second one, called *membershipValuesSynonymSet*, is a list-valued attribute where each of the elements in the list has a floating point value within [0–1] holding the relevance of the corresponding synonym in the value of the *synonymSet* list, to the domain of the ontology.

We have used the Wikipedia disambiguation pages corresponding to applicable concepts to calculate the values for the newly added *membershipValueLabel* and *membershipValuesSynonymSet* attributes. We have used the *reciprocal rank* (or, *first answer reciprocal rank*) commonly employed for the evaluation of question answering systems [12, 13] as the values of these attributes. If the system's response is at rank_i in the ranked list of answers, then its *reciprocal rank* is calculated as $1/rank_i$.

The formal definitions of these newly added fuzzy attributes of PQONT and WONT are provided below:

Definition 1 The fuzzy ontology attribute *membershipValueLabel* corresponding to the *label* (l) attribute of a concept in an ontology is a membership function $\mu(l)$ defined as follows, where $wiki_d(l)$ denotes the Wikipedia disambiguation page for l and $wiki_a(l)$ denotes the Wikipedia article page for l exactly describing the ontological sense of l:

$$\mu(l) = \begin{cases} 1.0 & \text{if neither } wiki_d(l) \text{ nor } wiki_a(l) \text{ exists, or only } wiki_a(l) \text{ exists} \\ 1/n & \text{if the use matching the ontological sense of } l \text{ is ranked } n\text{th} \\ & \quad \text{in } wiki_d(l) \\ 0.0 & \text{otherwise} \end{cases}$$

Definition 2 The list-valued fuzzy ontology attribute *membershipValuesSynonymSet* corresponding to the list-valued *synonymSet* ($ss(l) = \{s_1, s_2, \ldots, s_n\}$) attribute of a concept in an ontology is a membership function $\omega(ss)$, which returns the list of $\mu(s_i)$ (defined above) values for each synonym s_i of the concept:

$$\omega(ss) = \{\mu(s_1), \mu(s_2), \ldots, \mu(s_n)\}$$

To illustrate the employment of the *reciprocal rank* within our settings, we can consider the *Power* concept of PQONT. As the *label* value of the *Power* concept of PQONT is *power*, we first check the corresponding Wikipedia disambiguation page at http://en.wikipedia.org/wiki/Power_(disambiguation). Among the uses of the term *power* within this page, the one corresponding to its definition within the power quality domain is the fourth one, so its *reciprocal rank* is $1/4 = 0.25$ which in turn is used as the value of the *membershipValueLabel* of the *Power* concept. Following this procedure, the *reciprocal ranks* for the synonyms of the ontology concepts are calculated and used to form the lists of values to fill in the *membershipValuesSynonymSet* attributes. For those labels or synonyms for which no disambiguation page exists or just a single Wikipedia article corresponding exactly to the label or synonym or no separate Wikipedia article exists, the corresponding fuzzy attributes take the value of 1.0. If the considered Wikipedia disambiguation page does not include any use

that can be associated with the label or the synonym of an ontology concept, then 0.0 is set as the value of the *membershipValueLabel* attribute or as the value of the corresponding element of the *membershipValuesSynonymSet* attribute of the concept under consideration.

The final forms of PQONT and WONT which have been extended as described above to facilitate text processing applications are made publicly available for research purposes at http://www.ceng.metu.edu.tr/~e120329/FPQONT.owl and http://www.ceng.metu.edu.tr/~e120329/FWONT.owl, respectively.

Prospective text analysis applications of these extended ontologies include the following:

- Text categorization systems which can classify documents as related to the energy domain, or more specifically, as related to power quality or wind energy domains.
- Search and analysis of social media texts such as tweets to determine those ones related to the domains of the ultimate ontologies.
- Searching the Web (such as news sources) for texts regarding the domains of the ontologies in order to build semantic portals for these domains.

In all of the above applications, the new membership holding attributes of the ontology concepts can be used together with a convenient threshold value in such a way that if the sum of the memberships of the appearing related phrases in a document (i.e., a Web page, a news article, a tweet, etc.) exceeds the threshold, that document is considered as relevant to the domain of the ontology under consideration.

In order to illustrate the utilization of the ontologies when processing news texts or social media texts, we consider using the news video transcription datasets employed in [14] comprising 522 distinct news stories and a tweet dataset of about 200 K tweets randomly sampled through 28 consecutive days. Yet, although there are very few energy-related individual items (less than 1 %) in both datasets, there are no items particularly regarding the power quality or wind energy domains, due to the fact that FPQONT and FWONT are ontologies proposed for very specific energy domains and far larger datasets are required to illustrate their employment in these settings. Hence, instead of the second and third items above, we target at developing a text categorization system for scholarly articles by employing FPQONT and we describe this system with its evaluation results in the following section.

4 Categorizing Scholarly Articles Using the Extended Power Quality Ontology

Automatic text categorization (or, classification) systems are important tools due to the abundance of available textual resources. By means of these systems, people are able to retrieve the documents of their interests instead of being exposed to large sets of documents on a wide range of different topics.

We have built a simplistic text categorizer, based on the extended power quality ontology (FPQONT), which classifies scholarly articles as *relevant* to the domain or

not. It performs this classification by using the values of the *label* and *synonymSet* attributes of the ontology concepts as domain terms and the corresponding values of the fuzzy attributes of *membershipValueLabel* and *membershipValuesSynonymSet* as weights, so that a *relevance score* is calculated for each article, using the formula below.

$$relevance\ score = \sum_{i=0}^{c} \left[freq(l_i) * \mu(l_i) + \sum_{j=0}^{ss_i} freq(s_{i,j}) * \mu(s_{i,j}) \right]$$

Within the formula, c is the total number of concepts, l_i denotes the *label* value of ith concept, $\mu(l_i)$ is the value of the *membershipValueLabel* of the th concept, ss_i is the number of synonyms for the ith concept (e.g., the size of the value of its *synonymSet* attribute), $s_{i,j}$ is the jth element within the value of *synonymSet* of the ith concept, $\mu(s_{i,j})$ is the jth element within the value of *membershipValuesSynonymSet* of the ith concept (e.g., the membership value of the jth synonym to the corresponding concept), and finally the function $freq(x)$ denotes the frequency of the phrase (a label or synonym value), x, within the article under consideration.

After the calculation of the *relevance score* for an input article, if it is above a threshold, then the FPQONT-based categorizer classifies the article as *relevant*, otherwise the article is classified as *not-relevant*. Within the course of this preliminary text categorization application, we take the threshold as 1.0 and finding an optimal value for this threshold is left as future work.

As our target domain is that of electrical power quality, we have created a proprietary dataset of 10 relevant and 100 not-relevant journal articles, which are published in *IEEE Xplore Digital Library* site (http://ieeexplore.ieee.org/Xplore/home.jsp). This dataset comprises only the publicly-available titles and the abstract sections of these 110 articles and has about 17,700 words. Below, we outline our procedure to compile this dataset:

- We have compiled the titles and abstracts of 10 relevant articles which are highly cited papers published in the following journals, where the number of papers included from each journal is given in parentheses: *IEEE Trans. Ind. Appl.* (1), *IEEE Trans. Instrum. Meas.* (1), *IEEE Trans. Power Del.* (5), and *IEEE Trans. Power Electron.* (3).[3]
- For the 100 not-relevant articles, we have compiled sets of 10 articles each from 10 research domains distinct from the domain of electrical power quality. We have determined these research domains loosely based on the popular categories of the *WikiCFP site for call for papers* (http://www.wikicfp.com/cfp/allcat). Hence, we have compiled 10 mostly-cited journal papers from the following research domains, where the names of the source journals are given in parentheses: biomedical engineering (*IEEE Trans. Biomed. Eng.*), data and knowledge engineering (*IEEE Trans. Knowl. Data Eng.*), education (*IEEE Trans. Educ.*), fuzzy

[3] The journal names are given as denoted in the *Reference Abbreviation* column of the related IEEE document available at http://www.ieee.org/documents/tjmnames.pdf.

Table 1 Performance evaluation results of the baseline and FPQONT-based text categorizers

	True positives	False negatives	True negatives	False positives	Accuracy
Baseline categorizer	10	0	86	14	87.27
FPQONT-based categorizer	10	0	99	1	99.09

systems (*IEEE Trans. Fuzzy Syst.*), multimedia (*IEEE Trans. Multimedia*), nanotechnology (*IEEE Trans. Nanotechnol.*), networking (*IEEE/ACM Trans. Netw.*), robotics (*IEEE Trans. Robot.*), software engineering (*IEEE Trans. Softw. Eng.*), and speech/language processing (*IEEE Audio, Speech, Language Process.*).

In addition to the implementation of this FPQONT-based text categorizer, we have implemented a baseline categorizer which instead takes all membership values (both for labels and synonyms) as 1.0 during the calculation of the aforementioned *relevance score*, for comparative purposes. Like the FPQONT-based categorizer, the baseline categorizer also takes the threshold value as 1.0. The performance evaluation results of these ultimate categorizers are provided in Table 1, with breakdowns of their outputs (in columns 2–5) and their overall accuracies (in the last column).

The evaluation results demonstrate that both the baseline and FPQONT-based categorizers perform good at classifying truly relevant articles as *relevant*, as they both correctly categorize all 10 articles on electrical power quality as *relevant*. For these articles, the baseline categorizer calculates far higher *relevance score* values (ranging between 7.0 and 17.0) compared to those calculated by the FPQONT-based categorizer (ranging between 1.5 and 8.084), as expected. Considering the 100 not-relevant articles in our dataset, the baseline system incorrectly classifies 14 of them as *relevant* while the FPQONT-based recognizer only misclassifies one of them. Though our dataset of scholarly articles is limited in size, the FPQONT-based categorizer achieves an improvement of about 13.5 % over the baseline categorizer. Thereby, the proposed categorizer stands as a significant application example demonstrating the contribution of the proposed fuzzy extensions to the energy ontologies. Future work includes testing the proposed categorizer on larger datasets with different threshold values and relevance score formulae, and implementing other text processing applications which can benefit from the proposed ontologies.

5 Conclusion

In this paper, we first present our motivation for fuzzy extensions to two energy-related domain ontologies, namely, one for electrical power quality and one for wind energy, particularly to facilitate their employment in text analysis applications. Then, we describe our fuzzy extension procedure which is mainly based on the Wikipedia

disambiguation pages corresponding to the ontology concept labels and their synonyms. The ultimate power quality and wind energy ontologies are made publicly available for research purposes. Plausible text processing applications for which these extended ontologies can be more beneficial compared to their original crisp versions include text categorization systems, social media analysis applications, and semantic portals. A text categorization system for scholarly articles in the domain of electrical power quality is implemented as an application example and its performance is evaluated on a dataset of articles.

References

1. A. Gómez-Pérez, M. Fernández-López, O. Corcho, *Ontological Engineering*, 3rd edn. (Springer, New York, 2004)
2. R. Stevens, C. Goble, I. Horrocks, S. Bechhofer, Building a bioinformatics ontology using OIL. IEEE Trans. Inf Technol. Biomed. **6**, 135–141 (2002)
3. J. Morbach, A. Yang, W. Marquardt, OntoCAPE-a large-scale ontology for chemical process engineering. Eng. Appl. Artif. Intell. **20**, 147–161 (2007)
4. D. Küçük, O. Salor, T. İnan, I. Çadırcı, M. Ermiş, PQONT: a domain ontology for electrical power quality. Adv. Eng. Inform. **24**, 84–95 (2010)
5. D. Küçük, Y. Arslan, Semi-automatic construction of a domain ontology for wind energy using Wikipedia articles. Renewable Energy **62**, 484–489 (2014)
6. S. Grimm, A. Abecker, J. Völker, R. Studer, Handbook of Semantic Web Technologies, in *Ontologies and the Semantic Web*, ed. by J. Domingue, D. Fenseland, J.A. Hendler (Springer, New York, 2011), pp. 507–580
7. D. Parry: A Fuzzy Ontology for Medical Document Retrieval. In: Proceedings of the Australasian Workshop on Data Mining and Web Intelligence, (2004)
8. C.S. Lee, Z.W. Jian, L.K. Huang, A fuzzy ontology and its application to news summarization. IEEE Trans. Syst. Man Cybern. B Cybern.-Part B: Cybernetics **35**, 859–880 (2005)
9. N. Xue, S. Jia, J. Hao, Q. Wang, Scientific ontology construction based on interval valued fuzzy theory under Web 2.0. J. Software **8**, 1835–1842 (2013)
10. F. Li, W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, P. Zhang, Smart transmission grid: vision and framework. IEEE Trans. Smart Grid **1**, 168–177 (2010)
11. Smart Grid—Wikipedia. (http://en.wikipedia.org/wiki/Smart_grid)
12. E.M. Voorhees, D.M. Tice, The TREC-8 Question Answering Track Evaluation. In: Eighth Text REtrieval Conference (TREC-8), (1999)
13. D.R. Radev, H. Qi, H. Wu, W. Fan, Evaluating Web-based Question Answering Systems. In: Language Resources and Evaluation Conference, (2002)
14. D. Küçük, A. Yazıcı, Exploiting information extraction techniques for automatic semantic video indexing with an application to Turkish news videos. Knowl.-Based Syst. **24**, 844–857 (2011)

Evaluating Quality of Dispersion Based Fixation Detection Algorithm

Katarzyna Harężlak and Paweł Kasprowski

Abstract Information hidden in the eye movement signal can be a valuable source of knowledge about a human mind. This information is commonly used in multiple fields of interests like psychology, medicine, business, advertising, or even software developing. The proper analysis of the eye movement signal requires its elements to be extracted. The most important ones are fixations—moments when eyes are almost stable and the brain is acquiring information about the scene. There were several algorithms, aiming at detecting fixations, developed. The studies presented in this paper focused one of the most common dispersion-based algorithms—I-DT one. The various ways of evaluating its results were analyzed and compared. Some extensions in this algorithm were made as well.

1 Introduction

Human eyes play important role in the interpersonal communication and gathering knowledge regarding surrounding world. The desire to understand this learning process leads to asking many questions: What is a subject looking at? What does one see looking at a given point? Did one find searched information? What kind of information was gained when looking at a particular area? Is one looking at expected point of regard? Finding answers to those and other questions is an important task in many fields of interests like psychology, medicine, business, advertising, or software developing. This need is reflected in current research areas, among which the cognizance of an eye movement signal has a significant place, because information hidden in this signal can be a valuable source of knowledge.

Studies conducted in this field resulted in distinguishing few components of the signal. Its fundamental unit is a fixation when point of gaze remains within a small area for a given time. Fixations are interlaced with saccades—quick movements

K. Harężlak (✉) · P. Kasprowski
Silesian University of Technology, Institute of Informatics, Gliwice, Poland
e-mail: katarzyna.harezlak@polsl.pl

P. Kasprowski
e-mail: kasprowski@polsl.pl

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_11

made to reach another point of regard [2, 6, 7, 11, 16]. The example interpretations of fixations and saccades features, in terms of their usability in a cognitive process of human behaviors, were presented in the [10]. These guidelines show how important is a precise separation of these parts from an eye movement signal.

However, deeper analysis of a fixation reveals, within it, another types of movement: tremors, microsaccades, and drifts [7]. A quality of measurements is also an important issue. For this reason, the fixation cannot be thought as one point and some additional measures have to be involved like, e.g., a size of spatial dispersion between points in the fixation. Additionally, an eye movement signal differs in a characteristic for a particular subject and task, which makes the identification of fixations a complex task, still being solved by researches.

There are several algorithms developed for identifying fixations and saccades. Among them Dispersion-Based and Velocity-Based algorithms are the most popular ones [12, 13]. The first group of methods identifies fixations analyzing distances of the consecutive points. Group of points satisfying the condition defined by a given dispersion threshold is treated as a fixation. This threshold can refer to various measures—a distance between any two points, the largest distance between points in a fixation, or the largest distance from the center of a fixation to one of constituting its points [12, 13]. The most often analyzed algorithms in this group are I-DT (Dispersion Threshold Identification) and MST (Minimum Spanning Tree Identification) [5, 8, 12, 13].

The studies of points velocity entitled development of algorithms separating fixations and saccades points, taking their point-to-point velocities into account. Well-known representatives of these methods are (I-VT) Velocity Threshold Identification and (HHM) Hidden Markov Model Identification methods [5, 8, 12, 13].

The common goal of these algorithms is to extract sets of fixations and saccades facilitating the interpretation of eye movement signals. Fixations are usually considered to have a duration of at least 150 ms. [6, 12, 13], nevertheless discussions concerning a fixation duration in terms of performed tasks can be found [11, 15]. However, there are other factors influencing the analyzed measure. The outcome depends on input parameters, which are different kinds of thresholds. Various range of these parameters' values can lead to diversity of results, i.e., a number and duration of fixations and saccades. For this reason, several experiments were conducted to check these parameters' impact on the obtained results [1, 2, 5, 12].

Although the aforementioned studies let for drawing interesting conclusions, they did not exhaust the topic. The aim of the research is to continue the analysis of the I-DT dispersion-based algorithm, which is claimed to be robust one [2, 12] taking fixation identification into account.

2 Algorithm Description

The presented studies involved, as a basis, classic I-DT algorithm [12], converting eye movements signal into list of fixations in two steps. In the first step, each gaze point is classified as a fixation (F) or a saccade (S). The point is considered to be a part

of a fixation when some amount of previous points (called *window*) are closer than a predefined threshold. There are different possibilities how to measure the distance [1, 12, 13] but the most common is the maximal distance between any two points in the window.

The second step is a consolidation of F-points, laying one by one, into fixations. Fixations with length lower than another predefined threshold (called later *minLength*) are removed. The output of the algorithm is a set of fixations. Every fixation has four parameters: *start*, *length*, and *position* with two values: *x*, *y*. Parameters *x* and *y* are typically calculated as mean of all points in a fixation although there are other possibilities (like, e.g., median).

The algorithm used in this research extended a classic I-DT algorithm by introducing an additional step that makes it more flexible for low quality data (similar to [5]). The classic I-DT algorithm builds fixations list in step 2 based on *minLength* threshold—that is, forms fixations from fixation points courses that are longer than that threshold. In our modified algorithm every course of F points is used to build a fixation. In additional step 3, we calculate spatial and temporal distances between every two neighboring fixations. If both distances are below the threshold, the two fixations are merged into one. Only after this step *minLength* threshold is applied to every fixation.

Dispersion threshold algorithm
Input: list of (x,y) points
Step 1
Classify each point as a fixation (F) or a saccade (S) based on history:
Point is a fixation when max distance among window previous points is less than threshold
Step 2
Build fixations from groups of neighboring F points:
For every course of at least two subsequent F points
build fixation(start,length,x,y) where x and y are average values of x and y
among all points in a fixation.
Step 3
Merge neighboring fixations:
For every two subsequent fixations:
if temporal gap (saccade length) between them is less than *tgapThres*
and spatial gap (Euclidean distance between them) is less than *sgapThres*
then merge these two fixations into one.
Step 4
Remove too short fixations:
For every fixation:
if a fixation length is less than minLength remove the fixation.
Output/result: list of fixations

The additional fixations merging step should make the algorithm more robust to artifacts—sudden and unexpected changes of measured eye position due to imperfection of eye tracker algorithms and noise. Artifacts removal in an eye movement signal is not simple, especially when one does not want to affect parts of a correct signal. Similar setup, but for I-VT algorithm and restricted only for 'trackloss' situation (when there is no data available) was presented in [8].

3 Quality Evaluation

The main problem for every fixation detection algorithm is how to check whether it works correctly. The ground truth is in most cases not possible to determine as we don't know the correct positions of fixations and saccades in an eye movement signal. There are multiple ways to solve the problem. References [8] and [14] compared fixations detected by their algorithm to fixations detected by manual inspection made by experienced users. Reference [1] used somewhat arguable similar criterion: if different algorithms give similar data the data is considered to be reliable. The other possible criterion proposed in [3] is the calculation of so-called *nearest neighbor index* (NNI) for every fixation.

In this study we used a specially designed stimulus with a point jumping over a screen. The task of a person being tested was to follow, with eyes, the point, which was changing its location in specific moments of time. The fact that fixation placements were known, gave us opportunity to create an estimated correct sequence of fixations (later called *template sequence*) and compare the results of the algorithm to this sequence.

Experiment. There were altogether 24 participants taking part in the experiments, for which 40 recordings were registered using the stimulus described earlier. To obtain meaningful results there were only samples with accuracy error, estimated during calibration step, lower than 1.5 deg. chosen. Every stimulus presentation consisted of 21 points evenly distributed over the whole screen. The point was displayed in every location for about 3 s. Eye movements were recorded with the eye tracker using single web camera with USB 2.0 interface, the sampling frequency was 20 Hz.

Methodology. The algorithm presented in Sect. 2 was used to produce sequences of fixations for every sample. The algorithm was started with different values of *threshold*, spatial gap (*sgapThres*) and temporal gap (*tgapThres*). Parameters *window* and *minLength* were set to 5 as it seems to be a reasonable choice according to the literature. Assessment of the obtained results—sequences of fixations generated by the algorithm with various values of three parameters—*threshold*, *sgapThres* and *tgapThres*—was done using several metrics described in Table 1 To provide the correctness of the algorithm, metrics described in Table 1, were calculated for every set of parameters and compared to metrics calculated for template fixation sequence. For that sequence the values were: AFN = 21, AFD = 71.29, ASN = 20, and ASA = 19.2.

Additionally, there were several metrics calculated—presented in Table 2—that directly measured difference between a given sequence and the template sequence. More information about metrics used may be found in [4] and [5].

4 Results

At the beginning of the studies an influence of one parameter—*threshold*—on the results provided by the algorithm was checked. Its value initially set to 0.5 degree, was incremented by 0.05 up to the value of 10 degrees. The *sgapThres* and *tgapThres*

Table 1 Metrics used to describe sequence of fixations

AFN—Average fixation number	Amount of fixations divided by a number of elements in an analyzed set of values
AFD—Average fixation duration	A value of summarized fixations length, measured in milliseconds, divided by a number of elements in an analyzed set of values
ASN—Average saccades number	Amount of saccades divided by a number of elements in an analyzed set of values
ASA—Average saccades amplitude	Sum of distances between every two fixations divided by a number of elements in an analyzed set of values.

Table 2 Metrics used to calculate the similarity to the template

FQnS—Fixation quantitative score	Percentage of points included into a fixation, for which distance from a stimulus position is less than third part of a last saccade amplitude.
FQlS—Fixation qualitative score	A value of summarized distances between calculated fixations and stimuli positions divided by a number of stimuli positions presented.
SQnS—saccade quantitative score	A value of summarized saccades amplitudes divided by summarized distances between stimuli positions presented
LevDist—levenstein distance	A Levenshtein distance between a calculated sequence of fixations and the template sequence

were constant with value 0. As a main metrics, assessing a quality of the result, Levenstein distance (*LevDist*) was chosen. The analysis of its values revealed that there are two threshold ranges having meaningful influence on them. Low thresholds values caused splitting a fixation, when an amplitude of eyes trembling during a fixation was higher than a given threshold. For the described studies, it was a case when threshold was lower than 2.5 deg., which can be observed in the Fig. 1 (left side). Above this value the stabilization of *LevDist* was noticed as far as the threshold reached the second range, values higher than 8 degrees. Defining this parameter on such high level resulted in merging fixation and points of neighboring saccades, which entitled increasing of *LevDist* values.

To check the correctness of these findings they were compared to values obtained for *FQnS* metrics, which are presented in the Fig. 1 (right side). The symmetric shape of charts suggested the strong correlation between elements of both sets. This correlation turned to be full one with the coefficient equal -0.967. Similar relationship was found between *LevDist* and *ANF* metrics. In this case the correlation coefficient was -0.981

The analysis of the obtained results regarded a duration of a fixation as well. As it was mentioned earlier, the template of a fixation length was defined as time period, when a stimulus was displayed in one position (3,565 ms). It is a well-known problem that this value cannot be reached because of the fact that, when a point on the screen

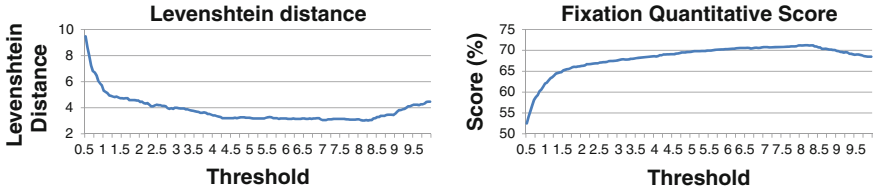


Fig. 1 Charts presenting the average values of *LevDist* (left side) and of *FQnS* (right side) metrics for various thresholds

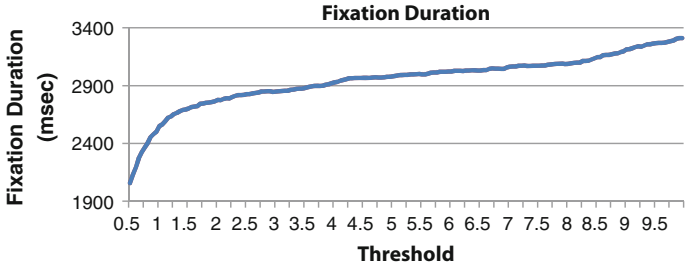


Fig. 2 Chart presenting the average values of *AFD* metric for various thresholds

changes its position, it takes some time for the human brain to react and to initiate an eye movement. For this reason duration of a measured fixation will never be ideal.

In Fig. 2 it can be noticed that in case of low threshold values, fixations found by the algorithm, feature by a short duration. It is another confirmation, that setting threshold in this scope can result in splitting one fixation into few small ones. It can also be observed that extending threshold to 10 degrees makes the values of *AFD* metrics to be closer to the ideal value. However, it cannot be assumed that for high threshold better results are obtained. As it was discussed above, this occurrence is the effect of attaching to fixation points, which in fact do not belong to it. This proves that the correctness of the found fixations should be ensured by more than one metrics. Last metrics taken during the studies into account were *ASA* and *SQnS* ones. Comparing these two sets of values, it turned out that their correlation is significant (with coefficient equal $= -0.542$) but not in such high level like in earlier described cases. Both of the metrics involve an amplitude of saccades but using it in slightly different way. First of them is strongly dependent on the number of fixations defined by the algorithm, which determines the number of saccades (ASN). The second metric is the ratio being the sum of found fixations amplitudes divided by the sum of saccades amplitudes existing in the template.

These various approaches entitle the differences in the results. In case of small threshold values a signal is divided into small fixations, between which saccades with amplitudes smaller than expected are defined. The sum of these amplitudes divided by the number of saccades provides low *ASA* value [Fig. 3 (left side)]. In case of the second metric (SQnS), the sum of such determined amplitudes can be in fact

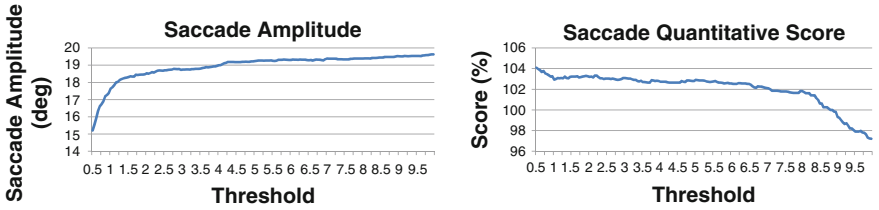


Fig. 3 Charts presenting the average values of *ASA* (left side) and *SQnS* (right side) metrics for various thresholds

almost equal to the sum of bigger amplitudes of the smaller set of saccades. This is why the *SQnS* value are almost stable, up to the threshold value of 8 degree, when the problem of merging a fixation with some points of neighboring saccades occurs [Fig. 3 (right side)].

5 Conclusion

The main goal of the research presented in the paper was to check how various parameters of well-known I-DT algorithm, used for extracting the set of fixations from the eye movement signal can influence the obtained results. The experiments are based on the specially designed stimulus with a point jumping over a screen. The known positions of the point were used as the reference template. Owing to that it was possible to determine, which ranges of parameters values provide reliable outcomes. It especially was visible for a main *threshold* parameter. The assessment was supported by the usage of various metric. Convergent results of independently calculated metrics values confirmed correctness of the algorithm. The results for the other parameters were not so unambiguous and need further studies. The average best *LevDist* for I-DT algorithm without additional merging step (i.e., *tgapThres* and *sgapThres* equal to 0) was 3.175. The average *LevDist* for the algorithm with the additional step was 2.85. Merging algorithm improved results in 28 % of cases. However, these results were achieved for different values of *sgapThres* and *tgapThres* optimized separately for every file. It was impossible to find one universal set of *sgapThres* and *sgapThres* values that on average for every file gave results better than without merging step (*sgapThres*=0, *tgapThres*=0). Nevertheless, it was observed that for the given *threshold* parameter value higher *tgapThres* thresholds improved the *LevDist* decreasing its value.

References

1. P. Blignaut, Fixation identification: the optimum threshold for a dispersion algorithm. Atten. Percept. Psychophysics **71**(4), 881–895 (2009)
2. P. Blignaut, T. Beelders, The effect of fixational eye movements on fixation identification with a dispersion-based fixation detection algorithm. J. Eye Mov. Res. **2**((5) (4)), 1–14 (2009)

3. M. Camilli et al., ASTEF: a simple tool for examining fixations. Behav. Res. Methods **40**(2), 373–382 (2008)
4. P. Kasprowski, O. V. Komogortsev, A. Karpov: First eye movement verification and identification competition at BTAS 2012. In The IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS 2012), 195–202 (2012)
5. Oleg V. Komogortsev, D.V. Gobert, S. Jayarathna, D.H. Koh, S.M. Gowda, Standardization of automated analyses of oculomotor fixation and saccadic behaviors. IEEE Trans. Biomed. Eng. **57**(11), 2635–2645 (2010)
6. B.R. Manor, E. Gordon, Defining the temporal threshold for ocular fixation in free-viewing visuocognitive tasks. J. Neurosci. Methods **128**(1–2), 85–93 (2003)
7. S. Martinez-Conde, S.L. Macknik, D.H. Hubel, The role of fixational eye movements in visual perception. Nat. Rev. Neurosci. **5**, 229–240 (2004)
8. S. M. Munn, L. Stefano, J. B. Pelz: Fixation-identification in dynamic scenes: Comparing an automated algorithm to manual coding. In: Proceedings of the 5th symposium on Applied perception in graphics and visualization, 33–42, ACM Press, (2008)
9. M. Nystrom, K. Holmqvist, An adaptive algorithm for fixation, saccade, and glissade detection in eye-tracking data. Behav. Res. Methods **42**(1), 188–204 (2010)
10. A. Poole, L.J. Ball, A. Poole, L.J. Ball, *Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects* (IGI Global, Hershey, 2005). In Encyclopedia of Human Computer Interaction
11. K. Rayner, Eye movements in reading and information processing: 20 years of research. Psychol. Bull. **124**(3), 372–422 (1998)
12. D. D. Salvucci, J. H. Goldberg: Identifying fixations and saccades in eye-tracking protocols. In: Proceedings of the 2000 Symposiumon Eye Tracking Research and Applications, 71–78, New York: ACM Press (2000)
13. F. Shic, K. Chawarska, B. Scassellati: The incomplete fixation measure. In: Proceedings of the 2008 Symposium on Eye Tracking Research and Applications, 111–114, New York: ACM Press (2008)
14. E. Tafaj, G. Kasneci, W. Rosenstiel, M. Bogdan: Bayesian online clustering of eye movement data. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA 12, 285–288, New York: ACM Press (2012)
15. R. van der Lans, M. Wedel, R. Pieters, Defining eye-fixation sequences across individuals and tasks: the Binocular-individual Threshold (BIT) algorithm. Behav. Res. Methods **43**(1), 239–257 (2011)
16. G. Veneriz, P. Piuyz, P. Federighi, F. Rosiniz, A. Federicoz, A. Rufa: Eye Fixations Identification based on Statistical Analysis—Case study.In Cognitive Information Processing (CIP), 2010 2nd International Workshop on, 446–451, (2010)

An Evaluation of Iris Detection Methods for Real-Time Video Processing with Low-Cost Equipment

Andrey Kuehlkamp, Cristiano Roberto Franco and Eros Comunello

Abstract The purpose of this work is to accomplish a study aiming the construction of an eye tracking and iris detection system, based on images obtained from a low-cost webcam. The main objective of the paper is to conduct a comparison between three computer vision approaches for iris detection, trying to identify the more suitable method for application in the aforementioned low-cost eye tracking system. The methods which have achieved the best detection rates were the Projection and Thresholding, however, all of them offer possibilities for application in real-time processing and improvement.

1 Introduction

Eye gaze tracking, or EGT, consists in a number of techniques which aim to determine the direction of a person gaze [1]. These techniques are widely used in most areas, especially in studies in the areas of interface usability, psychology, and the creation of accessibility technologies for disabled people [2].

In order to achieve such purpose, many strategies have been proposed in the last years, and many of them even became commercial products. These products are in general precise and efficient, but the devices have an elevated cost, which can, sometimes, reach tens of thousands of dollars [3]. As a main drawback, many of these solutions make use of intrusive equipment, such as head-mounted cameras or even special contact lenses.

Nonintrusive techniques for EGT are generally based on the detections of the eyes position from video images, and are called Remote Eye Gaze Tracking (REGT) [1].

A. Kuehlkamp (✉)
West Santa Catarina University UNOESC Videira, Catarina, Brazil
e-mail: kuehlkamp@gmail.com

A. Kuehlkamp · R. Franco · E. Comunello
Itajai Valley University UNIVALI Itajaí, Itajai, Brazil
e-mail: franco.cr@gmail.com

E. Comunello
e-mail: eros.com@gmail.com

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_12

The process usually starts with the capture of the subject image through video cameras. A sequence of processes is applied to locate the subject face and eyes. Eye features are extracted so that the position of the iris or of the pupil can be estimated. Finally, the position of the identified points in the eyes is used to correlate with the gaze direction [2].

For the intents of this study, it is considered that a REGT system must be capable of estimating the subject eye gaze as from the eyes movement, that is, in real time. The definition of real time may vary according to the context where it is applied, but Kehtarnavaz and Gamadia [4] propose that "real time in the perception sense" is the concept used mainly to describe the interaction between a human and a computer. In that interaction, an approximately instantaneous response is generated from the device, for each input given by the human user. Under this perspective, the limit in which a system can be considered as "real time" is up to 100 ms, because this is fast enough to be imperceptible to a human user.

In this paper, implementations for three known methods are described. The OpenCV (Open Source Computer Vision) library was used for the construction of the software. The objective of this study is to support the construction of a REGT system that is able to work using low-resolution images, obtained from a low-cost, laptop embedded webcam. More specifically, the purpose of this work is to conduct a comparison between three simple computer vision techniques for iris detection.

2 Related Works

Even though only the REGT techniques are considered, a variety of options can be employed to achieve the desired objective. The methods focused in this work can be categorized into three main types: (a) the application of a circular Hough Transform (HT) [6]; (b) Light Intensity Projection [7]; and (c) Light Intensity Thresholding [8].

Eye Tracking Types—Among all the proposed technologies for eye tracking, this work focuses on the solutions that are based on images. According to some authors [9–11], these solutions can be divided into Infrared-Based and Vision-Based.

Vision-based eye tracking techniques use visible light spectrum images obtained by digital cameras, and therefore, need no ad-hoc hardware. However, a bigger computational effort could be necessary for the face and eyes detection. Such effort is justified by the fact that no need for special IR lighting, what prevents the glint to be used to ease the location of the eyes. This way, the task of locating the eyes must be based on other visual features of the eyes, which can have more subtle characteristics.

In [12], an approach employs several techniques to deal with problems such as head movement, lighting and the low-cost devices limitations. Their system uses a modified HT in order to find circles which represent the iris in the image. The eye corners are detected to obtain reference points to estimate the gaze direction. A similar system was proposed in [13], in which a technique called Circle Frequency Filter (CFF), is used to locate each iris.

Iris Locating Using the HT—The Hough Transform is a well-known method in computer vision, used to determine parameters for simple geometrical forms, like lines and circles in images. A more complete description of the HT can be found in [14]. A number of works have already used some version of the HT to detect human irises in images, such as [2, 6, 10, 15].

Intensity Projection—The projection technique consists in creating a graphical representation based on the sum of the intensity of the pixels in the rows or columns of an image. Frequently, it is used to rapidly analyze the structure of an image and isolate its components. Projections of an image are one-dimensional representations of the image contents, usually computed alongside to the axis of the coordinates [16]. The horizontal and vertical projections of an image $I(u, v)$ with $0 \leq u < M, 0 \leq v < N$, where M and N are, respectively, the number of rows and columns in an image, can be defined as:

$$P_{\text{hor}}(v_0) = \sum_{u=0}^{M-1} I(u, v_0) \quad \text{for} \quad 0 < v_0 < N \tag{1}$$

$$P_{\text{ver}}(u_0) = \sum_{v=0}^{N-1} I(u_0, v) \quad \text{for} \quad 0 < u_0 < M \tag{2}$$

It is possible to find different approaches that apply the projection technique, in several works, such as [7, 17, 18].

Light Intensity Thresholding—Light intensity thresholding is another well-known technique for object segmentation in computer vision. This method is a kind of quantization, in which the pixel values are separated into two classes, based on a threshold value [16]. Several references to the application of this method can be found, like in [19], as well as some applications more specific to the iris detection in images, such as [1, 8, 15].

3 Development

Using the OpenCV, a program was created in order to capture webcam images and process them in real time, according to the aforementioned definition. The processing must produce as result the contour of each eye region, as well as the contour of the iris estimated position, overlapping the original image. Figure 1 shows the desired result of the process: in (a), the original image is presented; in (b), the extraction of the eyes region; and in (c), identification of each eye region and the iris.

Eye Region Extraction—After obtaining the initial image, the first task is to identify the eyes region. This was achieved by means of the Haar Classifier embedded in the OpenCV library. The Haar Classifier is an algorithm that was initially designed for face detection; however it can be easily adapted for the detection of any rigid object [5].

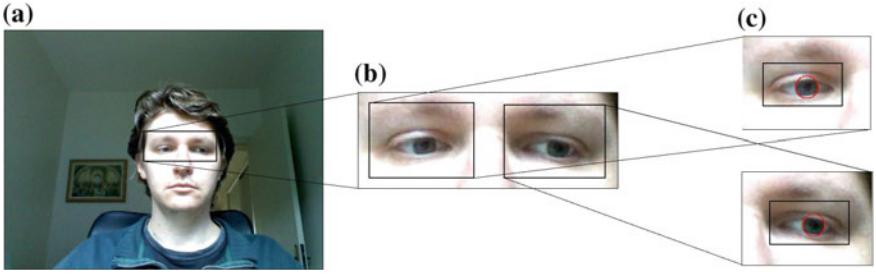


Fig. 1 **a** original image; **b** eye region detection; **c** processing expected result

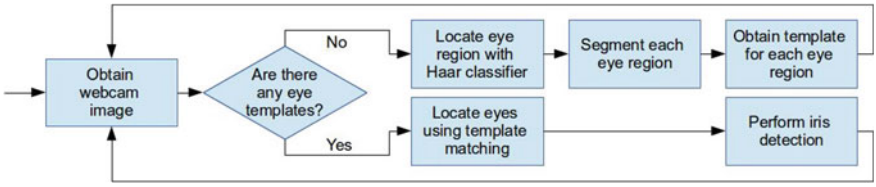


Fig. 2 Eye tracking process flowchart

The Haar Classifier may be used to locate faces or other objects in an image. It is based on the machine learning algorithm AdaBoost, and the OpenCV library provides several pretrained data sets for common objects detection. More details of this implementation may be found in [5]. An undesired result of the use of the Haar Classifier is that its performance may not be suited for real-time video processing. This way, it is employed only in order to obtain the eye templates, so that, with these, the eye tracking can be performed by another technique, the template matching. Figure 2 exhibits a flow diagram of this process.

The template matching consists in 'sliding' the template across the original image, and according to some comparison criterion, finding regions that are similar to the template [5]. After obtaining a region of the image which contains the eyes, the next step is to segment this region, so that each eye position is known. This was accomplished through the proportional division of the rectangular area which contains both eyes.

With the individual image of each eye as a template, it is no longer necessary to use the Haar Classifier. Once the eye region was delimited by the Haar Classifier, it is possible to narrow the template matching search, saving time and processing for the next video frames. From this point on, the initial processing of the original image was already done, so these processes will only be applied to the region where each eye was previously detected. Three methods were used to perform iris position detection in the image.

Hough Transform—The first iris detection method presented was based on Dobes et al. [6] and Torricelli et al. [2], who employed a modified version of the HT to

detect circles in images. The OpenCV library provides a version of the HT for circle detection, which was the version used in this work.

Intensity Projection—In the second method, the main idea was used by Fernandes Junior and Marengoni [7] and by Peng et al. [17], and it consists in the creation of a projection of the image, thus determining the highest intensity point in each axis. Normally, the iris region has a different level of intensity from the average of the image, so this method can be used to estimate its position.

Intensity Thresholding—The third method consists in using a global thresholding procedure, based on the work of Dong et al. [8], which tries to adjust the binarization threshold according to the average of the intensity in that region of the image.

The use of a static threshold in the binarization operation makes the procedure highly dependent on the illumination conditions and contents of the image. On the other hand, adaptive thresholding allows the discretization of details that are important to characterize the image, although local details are not so relevant for the purposes of this work. Hence, the chosen solution was the use of a global threshold, calculated over the average intensity of the image.

This method extracts the average intensity for this image, and based on this average, applies a binary thresholding. Erosion and opening morphological operations are applied to reduce undesirable noise. From the resulting binary image, the centroid of the binary object is calculated, which should correspond to the center of the iris. Based on a bounding box of the binary object, the radius of the iris circle is then estimated.

4 Experimental Environment

A comparison between these three methods was conducted, trying to identify the more suitable method for the construction of a low-cost eye tracking system. The criteria were defined as rates of: (a) iris detection hits; (b) wrong detections (false-positives); and (c) nondetection (misses). The evaluation was made individually to the left and right eyes, and later the average of these rates was calculated.

A number of image and video databases were considered, primarily the most used in facial computer vision research, such as UBIRIS.v2 [20], The University of Oulu Face Video Database [21], ZJU Eyeblink Database [22], The Yale Face Database B [12], Head Pose and Eye Gaze dataset [9].

In order to adequately evaluate the results of this work, as well as the future goals of this research, the following requirements should be fulfilled by data sets: (i) Data must be recorded in videos or image sequences (equal or higher to 15 fps); (ii) The data sets must provide ground truth information of the eye location in the images,in a manner that it is possible to compare the results obtained and measure their accuracy; (iii) The capture environment must be the closest possible to real scenarios, avoiding too much control in the environment (e.g., luminosity, background homogeneity). Among these databases, none of them achieved full compliance with the presented requirements, fact that motivated the creation of a specific data set.

Five video sequences were captured with a laptop's integrated 1.3 MP webcam. The laptop was equipped with an Intel Core 2 Duo 2.2 GHz and 8 GB RAM. As one of the goals is the application in low-cost equipment, the resolution used in the videos was 800×600 pixels, at a 30 fps rate. Three people were recorded, using different places and illuminations. No special illumination was used. Precautions were taken to keep the average processing time of the frames under 100 ms, so as to keep the performance within the range considered by the authors as real time.

The detection program was executed on each video, and the output was stored in a numbered sequence of image files, so that these could be later classified. A semiautomated method was used, where human evaluators could answer if the automated detection was performed correctly (the correct iris location was pointed—hit), mistakenly (false-positive, that is, a location that does not correspond to the iris was pointed), or there was no detection at all (miss). This way, the evaluators answer to the question for each image, and for each eye, using the following standard notation: Y—detection performed precisely; N—no detection; and F—mistaken detection.

5 Experimental Results

In each experiment performed, one video was recorded. Next, the program processed all the videos, generating a sequence of images with markers, for later classification. Table 1 shows the results obtained in the processing of all the video sequences, with all three detection algorithms.

In the first video, the best results were achieved with the HT and Thresholding algorithms. In this test a high rate of false-positives was generated by the Projection algorithm. This fact can be explained by the working principle of the algorithm. As it simply identifies a point in the image where the highest intensities occur, not taking into account other factors, it is possible that this point is identified in other region, not corresponding to the iris. In video sequences 2 and 4, the best results were obtained by the Projection algorithm, while the results of the HT and Thresholding algorithms were similar, both with high nondetection rates.

In the data collected from Video 3, some disparity in detection occurs through Projection method when each eye is considered separately. The same effect was observed in the fifth video sequence, with the HT algorithm. The same does not

Table 1 Iris detection results

Iris location															
	Hough					Projection					Thresholding				
Video #	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Hits	267	22	35	220	190	149	394	237	353	378	373	131	387	1	354
Misses	88	352	347	142	130	0	2	0	0	1	17	265	4	395	28
False-positive	41	22	14	34	76	247	0	159	43	17	6	0	5	0	14

Table 2 Overall performance of the iris detection methods

Iris location			
	Hough (%)	Projection (%)	Thresholding (%)
Hits	37.1	76.3	62.9
Misses	53.5	0.2	35.8
False-positive	9.4	23.5	1.3

occur with the Projection and Thresholding algorithms. Such variance may suggest a higher susceptibility of the algorithms to lighting variations between one side of the face and the other.

Data Evaluation—In a general average between the processed videos, the best detection rates were accomplished by the Projection technique (76.3 %), followed by Thresholding (62.9 %) and HT (37.1 %). On the other hand, the algorithm also presents the higher false-positive rate, with 23.5 % against 9.4 % for the HT and 1.3 % for the Thresholding. The highest nondetection rate was presented by the HT, with 53.5 %, against 35.8% for Thresholding and 0.2 % for Projection. Table 2 presents the overall result of this evaluation of the three implementations, applied to a total of 990 frames.

Among the presented results, the performances of the Projection and Thresholding are particularly evident, with hit rates above 60 %, or yet the low nondetection rate of the Projection (below 1 %).

6 Conclusion and Discussions

In spite the discrepancies in the iris detection rates, all three experimented techniques presented promising results, in specific situations. However, some considerations should be taken into account:

(i) Result differences between right and left eyes, indicates low tolerance to lighting variations; (ii) The detection techniques are conceptually different, and that explains the fact that the Projection technique presented such a low nondetection rate. This technique only points a place in the image where the intensity is distinct, and this place will occur in an image regardless if it contains an iris or not; (iii) The high nondetection rate of the HT and the Thresholding (between 35 and 55 %) is not uniform when the videos are individually analyzed. Historically, the Hough transform can generate high false-positive or nondetection rates under nonideal lighting conditions [10]. This may be interpreted as deficiencies in these algorithms to deal with some specific lighting conditions. (iv) Given the intention to evaluate methods for use in real time, only relatively simple techniques were employed. This way, all three methods had frame processing times under 30 ms, which enables their use in real time with up to 30 fps videos.

The Intensity Projection and Thresholding had the best detection rates. The features to be improved are directly connected with the susceptibility to environment

lighting factors, which were mainly responsible for the discrepancies between right and left eyes. Improvements in the detection program that is based on the HT must be made. Some studies obtained more successful outcomes using modified versions of the HT, such as [6] and [2], although under the additional cost of creating an ad-hoc implementation of this function. Several nondetection cases were caused by failure in the approximate eye region location, by the Haar Classifier. In these cases, the Haar classifier located regions that did not contain an iris to be detected.

An additional factor seems to have exerted some influence in the process are particular characteristics in the faces of the individuals, such as skin color, face format, and eye format and positioning. It is important to stress the need to create an adequate video database containing ground truth data to identify interest points in the faces. Such database will allow a better evaluation and comparation with similar works.

The main contribution of this work was to evaluate the proposed algorithms, in the sense of confirming their feasibility in the development of a real-time eye tracking technology based on regular computing devices. The study was also successful in pointing some deficiencies of each technique, which will be of great value to their future improvement, thus indicating the directions in the continuity of the research.

Acknowledgments This work was financially supported by FINEP (Financiadora de Estudos e Projetos), FAPESC (Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina), UNOESC (Universidade do Oeste de Santa Catarina), and the CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

References

1. C. Morimoto, M. Mimica, Eye gaze tracking techniques for interactive applications. Comput. Vis. Image Underst. **98**, 4–24 (2005)
2. D. Torricelli, S. Conforto, M. Schmid, T. DAlessio, A neural-based remote eye gaze tracker under natural head motion. Comput. Methods Programs Biomed. **92**(1), 66–78 (2008)
3. M. Kumar, Reducing the Cost of Eye Tracking Systems. Technical Report CSTR 2006–08, Stanford University, Stanford (2006)
4. N. Kehtarnavaz, M.N. Gamadia, *Real-Time Image and Video Processing: From Research to Reality* (Morgan & Claypool, US, 2006)
5. G. Bradski, A. Kaehler, *Learn. OPenCV* (OReilly, Sebastopol, 2008)
6. M. Dobes, J. Martinek, D. Skoupil, Z. Dobesova, J. Pospisil, Human eye localization using the modified Hough transform. Optik **117**, 468–473 (2006)
7. V. Fernandes Junior, M. Marengoni, Detecção e rastreamento de olhos para implementação de uma interface humano-computador, Anais V Workshop de Visão Computacional, Universidade Presbiteriana Mackenzie, São Paulo (2009)
8. W. Dong, Z. Sun, T. Tan, Z. Wei, Quality-based dynamic threshold for iris matching, in *ICIP* 2009, IEEE, pp. 1949–1952 (2009)
9. S. Asteriadis, D. Soufleros, K. Karpouzis, S. Kollias, A Natural Head Pose and Eye Gaze Dataset, in *ICMI*, Boston, 2–6 November, 2009
10. G. Crisafulli, G. Iannizzotto, F. La Rosa. Two competitive solutions to the problem of remote eye-tracking. 2nd Conference on Human System Interactions, 2009. pp. 356–362, IEEE (2009)
11. G. Daunys et al., Report on New Approaches to Eye Tracking. COGAIN, IST-2003-511598: Deliverable 5.2

12. A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intelligence **23**(6), 643–660 (2001)
13. M. Haseyama, C. Kaneko, A robust human-eye tracking method in video sequences. ICIP **2005**, 362–365 (2005)
14. M. Nixon, A. Aguado, *Feature Extraction & Image Processing*, 2nd edn. (Academic Press, New York, 2008)
15. Y. Chen, M. Adjouadi, C. Han, J. Wang, A. Barreto, N. Rishe, J. Andrian, A highly accurate and computationally efficient approach for unconstrained iris segmentation. Image Vis. Comput. **28**(2), 261–269 (2010)
16. W. Burger, M. Burge, *Digital Image Processing: An Algorithmic Introduction Using Java* (Springer, New York, 2008)
17. K. Peng, L. Chen, S. Ruan, G. Kukharev, A robust algorithm for eye detection on gray intensity face without spectacles. J. Comput Sci. Technol. **5**(3), 127–132 (2005)
18. M. Bianchini, L. Sarti, An eye detection system based on neural autoassociators. Artif. Neural Networks Pattern Recognit. **4087**, 244–252 (2006)
19. A.Z. Arifin, A. Asano, Image segmentation by histogram thresholding using hierarchical cluster analysis. Pattern. Recognit. Lett. **27**, 1515–1521 (2006)
20. G. Proença, S. Filipe, R. Santos, J. Oliveira, L.A. Alexandre, The UBIRIS.v2: a database of visible wavelength images captured on-the-move and at-a-distance. IEEE Trans. PAMI. **32**(8), 1529–1535 (2010)
21. B. Martinkauppi, X. Soriano, S. Huovinen , M. Laaksonen, Face video database. CGIV'2002, Poitiers, France, pp. 380–383 (2002)
22. G. Pan, L. Sun, Z. Wu, S Lao, Eyeblink-based Anti-spoofing in Face Recognition from a Generic Webcamera. ICCV'07, IEEE: Rio de Janeiro, Brazil, October 14–20, (2007)

Part III
Traffic Measurement and Analysis

Principles of Pervasive Cloud Monitoring

Gokce Gorbil, David Garcia Perez and Eduardo Huedo Cuesta

Abstract Accurate and fine-grained monitoring of dynamic and heterogeneous cloud resources is essential to the overall operation of the cloud. In this paper, we review the principles of pervasive cloud monitoring, and discuss the requirements of a pervasive monitoring solution needed to support proactive and autonomous management of cloud resources. This paper reviews existing monitoring solutions used by the industry and assesses their suitability to support pervasive monitoring. We find that the *collectd* daemon is a good candidate to form the basis of a lightweight monitoring agent that supports high resolution probing, but it needs to be supplemented by high-level interaction capabilities for pervasive monitoring.

Keywords Cloud monitoring · Pervasive monitoring · Cloud monitoring tools

1 Introduction

Cloud providers need to manage increasingly large and complex cloud infrastructures and assure the availability, reliability, elasticity, and performance of offered cloud services [1]. As the scale and complexity of cloud facilities increase, providers need to adopt an autonomous and proactive management approach in order to meet quality-of-service (QoS) guarantees at competitive prices [2]. While monitoring is

G. Gorbil (✉)
Department of Electrical and Electronic Engineering,
Intelligent Systems and Networks Group, Imperial College London,
London SW7 2AZ, UK
e-mail: g.gorbil@imperial.ac.uk

D. Garcia Perez
Atos Origin, Barcelona, Spain
e-mail: david.garciaperez@atos.net

E. Huedo Cuesta
Department of Computer Architecture and Automation,
Distributed Systems Architecture Group, Complutense University of Madrid,
28040 Madrid, Spain
e-mail: ehuedo@fdi.ucm.es

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_13

essential to many activities in the cloud [3], it is especially crucial for resource and performance management to enable autonomous control, which requires accurate and fine-grained monitoring of virtual [4] and physical computing and storage resources [5, 6], applications [7], and the network [8]. In this paper, we review the principles and requirements of pervasive cloud monitoring, and present how concepts from self-aware networks [9] can be leveraged to design a pervasive monitoring solution. We also provide a short review of existing cloud monitoring tools, and discuss integration possibilities with the OpenNebula cloud management platform.

2 Requirements of Pervasive Cloud Monitoring

Cloud monitoring needs to be elastic to accommodate for the dynamism of the cloud environment due to consumer churn and virtualization, and it needs to support runtime configuration changes. For autonomous and proactive cloud management, monitoring data is required to provide an accurate representation of the cloud state and needs to be delivered in a timely fashion in order to enable quick decisions in the face of changes. A pervasive monitoring solution needs to accommodate for volatility in virtual resources [4] and for the migration of virtual machines (VMs) [10]. The locations of the probes of the monitoring system that collect measurements from cloud resources are specified by the cloud layers given below [5], as the layer at which a probe is located limits the types of events and resources that the probe can observe and monitor [11, 12]. In this paper, we consider all layers except the facility layer as relevant to the monitoring solution.

- Facility layer: Consists of the physical infrastructure of the data center(s) where the cloud is hosted. Monitoring at this layer considers data center operations, energy consumption, environmental impact, and physical security, such as surveillance and architectural resilience [11].
- Network layer: Consists of the communication links and paths within a cloud, between clouds, and between the cloud and the user.
- Hardware layer: Consists of the physical components of the computing, storage and networking equipment.
- Operating system (OS) layer: Consists of the host and guest operating systems, and the hypervisors, i.e. the virtual machine (VM) managers.
- Middleware layer: Normally only present in the platform-as-a-service (PaaS) and software-as-a-service (SaaS) models, it consists of the software layer between the user application and the OS.
- Application layer: Consists of the user applications running in the cloud.
- User layer: Consists of the end users of the cloud applications and the applications running outside the cloud, e.g. a web browser of the end user.

Due to the complexity of cloud platforms and the heterogeneity of supported applications, there are many *metrics* of interest that need to be monitored concurrently [3]. In addition to actual measurements, all metrics can be evaluated in terms of

statistical indicators and temporal characteristics, which need to be supported by the monitoring solution in order to reduce the volume of monitoring traffic. Large-scale clouds present challenges in terms of the *scalability* of the monitoring solution due to the high number and types of resources that need to be monitored. Scalability is also important considering that the monitoring solution needs to be able to handle many metrics concurrently, perhaps as many as a hundred. The pervasive monitoring solution therefore needs to efficiently collect, transfer and analyze data from many probes without impairing the normal operations of the cloud. The scalability issue has been mainly addressed by aggregation and filtering of the monitoring data [6, 10, 13–16]. Other approaches to improve scalability include autonomous monitoring solutions that self-configure, for example by changing the monitoring intervals and parameters [2, 17]. In the next section, we discuss how concepts from self-aware networks can be adopted to perform *goal-oriented monitoring*, which addresses, among other things, the scalability issue.

3 Agent-Based Adaptive Pervasive Monitoring

One of the concerns in cloud monitoring is congestion and failures in the communication networks since any network disruptions affect not only the hosted cloud services but also the services used by the cloud provider for the monitoring and management of the cloud. Self-aware adaptive overlay networks [9, 18] provide an attractive solution to address these issues. We propose that an *inter-cloud overlay network* is constructed in order to monitor inter-cloud communications and to provide resilient and adaptive communications. In this section, we first present the Cognitive Packet Network (CPN) [19, 20], which is a self-aware packet routing protocol that implements many of the principles of pervasive monitoring in a network context in order to measure its performance and the conditions of the network in a distributed manner [21]. As the CPN self-monitors and self-manages the communication paths, it can autonomously adapt to changing conditions in the network. Its self-* properties make the CPN a good solution for the monitoring of intra-cloud and inter-cloud communications, and for autonomous adaptation of inter-cloud overlay paths. In addition to monitoring the network links and routers, CPN can also be used as the basis of a pervasive monitoring solution employed to collect measurements from cloud resources. We will present an overview of an adaptive agent-based pervasive monitoring solution based on the CPN later in this section.

Based on measurements it collects from the network nodes and links, CPN adaptively finds the best routes according to QoS criteria, such as low packet delay, specified by the users of the network. CPN employs adaptive learning techniques with random neural networks (RNNs) [22–24] and reinforcement learning [25] in order to make routing decisions at each network node. It uses smart packets (SPs) for exploring the network, dumb packets (DPs), which are source-routed, to carry the payload, and acknowledgments (ACKs) to collect measurements performed by the SPs and DPs and to train the neural networks. SPs are generated by the source node

on demand, i.e. when the user requests a new path with a given QoS goal or when it wants to discover parts of the network state. At each hop, SPs are routed based on the experiences of previous packets, employing a learning algorithm, mainly reinforcement learning on RNNs. The RNN is a neural network model inspired by biology, and it is characterized by positive (excitatory) and negative (inhibitory) signals in the form of spikes of unit amplitude which are exchanged between neurons and alter the potential of the neurons. A neuron is connected to one or more neurons, forming a neural network; each neural link has a weight, which can be positive or negative.

In the RNN, the state q_i of the ith neuron represents the probability that it is excited, and satisfies the following system of nonlinear equations:

$$q_i = \frac{\lambda^+(i)}{r(i) + \lambda^-(i)}$$

where

$$\lambda^+(i) = \sum_j q_j w_{ji}^+ + \Lambda_i^+, \quad \lambda^-(i) = \sum_j q_j w_{ji}^- + \Lambda_i^-, \quad r(i) = \sum_j w_{ij}^+ + w_{ij}^-$$

w_{ji}^+ is the rate at which neuron j sends excitation spikes to neuron i when j is excited, w_{ji}^- is the rate at which neuron j sends inhibition spikes to neuron i when j is excited, and $r(i)$ is the total firing rate of neuron i. Λ_i^+ and Λ_i^- are the constant rates of the external positive and negative signal arrivals at neuron i, respectively. These external signal arrivals follow stationary Poisson distributions. For an N neuron RNN, the parameters are the $N \times N$ weight matrices W^+ and W^- which need to be learned from the inputs.

Each node in the CPN stores an RNN for each QoS goal and source-destination pair. Each RNN has a neuron for each communication link at the node. The RNN is used to make adaptive decisions regarding the routing of SPs by routing the SPs probabilistically according to the weights of the neurons, which are updated using reinforcement learning. The QoS goal of the RNN is expressed as a function to be minimized, and the reward used in the reinforcement learning algorithm is simply the inverse of this function. Each received ACK triggers an update of the RNN based on the performance metrics observed by the SP or DP that resulted in the ACK.

We adopt the CPN in order to monitor cloud communications. In addition, we adopt concepts from the CPN to design an agent-based adaptive monitoring solution that performs goal-based monitoring, where metrics relevant to performance management are measured locally at each resource by *monitoring agents*, and collected on demand by *monitoring managers* to enable intelligent resource allocation decisions.

The use of configurable software agents has been proposed in the literature in order to construct a flexible monitoring architecture [26]. In agent-based monitoring solutions, agents located at the physical and virtual resources implement one or

more probes that collect measurements of metrics from the resource either on demand or periodically. Agent-based solutions are attractive for pervasive cloud monitoring since they provide elasticity, as the monitoring agents can be designed so that they start-up and shut-down with the resource they are attached to, e.g. VMs, and migrate with them, enabling easy monitoring of VMs even under migration. Furthermore, agents can be contacted at run-time in order to enable and disable monitoring activities on a resource-by-resource basis, providing a highly configurable and adaptive solution.

Monitoring agents store their measurements in local repositories. Smart packets are sent by the monitoring managers in order to collect measurements from these repositories, similar to the way that SPs are used in standard CPN by the source nodes. The monitoring managers make the collected measurements available to the resource management system to enable QoS-based decisions, and they may also configure monitoring activities of the agents depending on their QoS goals. It is expected that multiple monitoring managers will exist for fault tolerance and load balancing purposes.

4 A Review of Cloud Monitoring Tools

We provide a review of some of the popular cloud monitoring tools in Table 1; a more comprehensive review can be found in [27]. Out of the reviewed monitoring tools, Zabbix, Ganglia and Nagios are the most capable and flexible. They offer a single tool for the monitoring of different system and application metrics, enable the user to set-up and receive alerts and notifications, provide aggregation mechanisms for the monitored metrics, and keep a historical record of the monitoring data. All three projects are widely used and have strong community support.

Despite the capabilities of Zabbix, Ganglia and Nagios, we believe that *collectd* is the best candidate to form the basis of a lightweight monitoring agent as part of the pervasive monitoring solution outlined in the previous section, since it supports a wide variety of metrics and high resolution probing for many concurrent probes. collectd can also probe at different layers, e.g. at the application, middleware and OS layers, and thus can support both high-level and low-level monitoring. It is also customizable via plug-ins, allowing the addition of more application-specific probes. In order to add further capabilities to the pervasive monitoring solution, such as visualization and analysis, the system needs to be interfaced with high-level monitoring tools. Nagios appears to be the best choice for this purpose due to its extensibility and flexibility. Measurements from the collectd-based monitoring agents can be provided to Nagios via the collectd-nagios plug-in, and alerts for the user can be configured and transmitted by Nagios. Nagios would also present a front-end to the human user for visualization of metrics and configuration of the monitoring agents, which is possible by extending Nagios.

Table 1 Cloud monitoring tools

Tool	Description
Collectl	A lightweight monitoring tool that collects system-level information. Provides basic monitoring only, and limited to Linux systems
SAR	Offers basic system-level monitoring. Graphs supported via SAG
SIGAR	A portable API for system-level monitoring, providing bindings for Java, Python, Ruby, C#, etc. Not currently developed
Monit	A utility for monitoring and managing processes, programs, files, directories, and devices on Unix and Linux systems. Designed to conduct automatic maintenance and repairs
SFlow	Network traffic monitor. Widely supported by networking equipment, such as routers
Munin	Client-server based monitoring tool, designed for easy visualization of monitoring data. Mainly for Linux systems, but Windows systems are supported via third party plug-ins. Can be extended to perform actions based on the gathered data
Ganglia	A scalable distributed monitoring system developed for computing clusters and grids. Uses a multicast-based publish-subscribe protocol to decouple communication endpoints. Employs popular and proven technologies to represent (XML), transfer (XDR) and store monitoring data (RRDtool). Provides a graphical interface for visualization of monitoring data, in addition to an alert system. Extensible via custom clients
Nagios	A full monitoring solution designed to monitor applications, services, operating systems, network protocols and system metrics with a single tool. Supports configuration of actions to take when certain conditions are satisfied. Provides visualization and alerts
Zabbix	Monitoring solution mainly designed to monitor networks and network services. Uses SQL databases to store monitoring data, and provides a web front-end and an API to access it. System-level metrics on the hosts can be monitored via the provided agent
collectd	A lightweight daemon for periodically collecting system metrics, supporting a wide variety of metrics at different layers, e.g. application, OS, networking. Supports high resolution probing. Extensible via plug-ins. Provides several mechanisms to store the monitoring data, e.g. via RRDtool

5 Monitoring with OpenNebula

The autonomous cloud management system employs a cloud management toolkit that provides a “dumb” interface in order to actuate its decisions. The cloud community currently favors two open-source industry standard cloud management toolkits: OpenStack and OpenNebula. Both projects provide tools and services to set-up, configure and manage a cloud platform. At present, OpenNebula provides a more mature software stack, and easier installation and configuration.

OpenNebula relies on monitoring the cloud infrastructure in order to assess the state of the resources. OpenNebula’s monitoring subsystem gathers information from the physical hosts and the VMs by executing a set of static probes in the monitored resources, the output of which are sent to the cloud manager using either push or

pull mode. When using the *pull mode*, the manager periodically and actively queries each host. This mode is limited by the number of active connections that can be made concurrently since hosts are queried sequentially. As such it is only appropriate for small-scale clouds, consisting of 50 hosts or less, and when low update frequencies are acceptable (e.g. for 50 hosts, the monitoring period would typically be around 5 min). In the UDP-based *push mode*, each host periodically sends monitoring data to the manager. This is more scalable than the pull mode and therefore better suited for larger infrastructures and high update frequencies.

OpenNebula allows simple customization of the monitoring drivers and the creation of new drivers which exclusively use the pull model to report the data. OpenNebula also provides the *OneGate module* to enable VMs to push application-related monitoring information to the manager. However, OneGate is designed for only small-scale application-layer monitoring. The pervasive monitoring solution and OpenNebula can be integrated by either implementing a new OpenNebula monitoring driver in order to interface the monitoring agents directly with OpenNebula, or by integrating OpenNebula with a third-party tool such as Nagios or Ganglia that acts as an intermediary between the monitoring agents and OpenNebula.

6 Future Work

We provided an overview of an agent-based design for a pervasive monitoring solution based on concepts from self-aware networks, and reviewed some of the popular cloud monitoring tools. Among the tools reviewed, collectd is the best candidate to form the basis of a lightweight monitoring agent. In future work, we will extend our initial monitoring design, and evaluate the performance of resource allocation decisions based on goal-oriented monitoring.

Acknowledgments The work presented in this paper was partially supported by the EU FP7 research project PANACEA (Proactive Autonomous Management of Cloud Resources) under grant agreement no. 610764.

References

1. M. Armbrust, A. Fox, R. Griffth, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)
2. T. Lorimer and R. Sterritt, Autonomic management of cloud neighborhoods through pulse monitoring, in: *Proceedings of 5th IEEE International Conference on Utility and Cloud Computing (UCC'12)*, pp. 295–302, November 2012
3. G. Aceto, A. Botta, W. de Donato, A. Pescape, Cloud monitoring: a survey. Comput. Netw. **57**(9), 2093–2115 (2013)
4. F.-F. Han et al., Virtual resource monitoring in cloud computing. J. Shanghai Univ. (Engl. Ed.) **15**(5), 381–385 (2011)

5. J. Montes et al., GMonE: a complete approach to cloud monitoring. Future Gener. Comp. Syst. **29**(8), 2026–2040 (2013)
6. J. Povedano-Molina et al., DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant clouds. Future Gener. Comp. Syst. **29**(8), 2041–2056 (2013)
7. K. Alhamazani et al, Cloud monitoring for optimizing the QoS of hosted applications, in: *Proceedings of 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom'12)*, pp. 765–770, December 2012
8. L. Atzori, F. Granelli, A. Pescape, A network-oriented survey and open issues in cloud computing, *Cloud Computing: Methodology, Systems, and Applications* (CRC Press, Florida, 2011), pp. 91–108
9. E. Gelenbe, Steps toward self-aware networks. Commun. ACM **52**(7), 66–75 (2009)
10. B. Konig, C.J.M. Alcaraz, J. Kirschnick, Elastic monitoring framework for cloud infrastructures. IET Commun. **6**(10), 1306–1315 (2012)
11. J. Spring, Monitoring cloud computing by layer, part 1. IEEE Secur. Priv. **9**(2), 66–68 (2011)
12. J. Spring, Monitoring cloud computing by layer, part 2. IEEE Secur. Priv. **9**(3), 52–55 (2011)
13. Y. Meng, Z. Luan, Z. Cheng, and D. Qian, Differentiating data collection for cloud environment monitoring, in: Proceedings of 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM'13), pp. 868–871, May 2013
14. J.S. Ward and A. Baker, Monitoring large-scale cloud systems with layered gossip protocols, arXiv Computing Research Repository, vol. abs/1305.7403, May 2013
15. H.T. Kung, C.-K. Lin, and D. Vlah, CloudSense: Continuous fine-grain cloud monitoring with compressive sensing, in Proceedings of 3rd USENIX W'orkshop on Hot Topics in Cloud Computing (HotCloud'11), June 2011
16. C. Canali, R. Lancellotti, Improving scalability of cloud monitoring through PCA-based clustering of virtual machines. J. Comput. Sci. Technol. **29**(1), 38–52 (2014)
17. G. Katsaros et al., A self-adaptive hierarchical monitoring mechanism for clouds. J. Syst. Softw. **85**(5), 1029–1041 (2010)
18. R. Lent, O.H. Abdelrahman, G. Gorbil, A Low-Latency and Self-Adapting Application Layer Multicast, *Computer and Information Sciences* (Springer, Netherlands, 2010), pp. 169–172
19. E. Gelenbe, R. Lent, A. Nunez, Self-aware networks and QoS. Proc. IEEE **92**(9), 1478–1489 (2004)
20. E. Gelenbe, Z. Xu, E. Seref, Cognitive packet networks, in: Proceedings of 11th International Conference on Tools with Artificial Intelligence, pp. 47–54, November 1999
21. G. Sakellari, The cognitive packet network: a survey. Comp. J. **53**(3), 268–279 (2009)
22. E. Gelenbe, Sensible decisions based on QoS. Comput. Manag. Sci. **1**(1), 1–14 (2003)
23. E. Gelenbe, S. Timotheou, Random neural networks with synchronised interactions. Neural Comput. **20**(9), 2308–2324 (2008)
24. E. Gelenbe, K. Hussain, Learning in the multiple class random neural network. IEEE Trans. Neural Netw. **13**(6), 1257–1267 (2002)
25. U. Halici, Reinforcement learning with internal expectation for the random neural network. Eur. J. Oper. Res. **126**(2), 288–307 (2000)
26. R. Aversa, L. Tasquier, and S. Venticinque, Management of cloud infrastructures through agents, in: Proceedings of 3rd International Conference on Emerging Intelligent Data and Web Technologies (EIDWT'12), pp. 46–52, Sep. 2012
27. K. Alhamazani et al. An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art, arXiv Computing Research Repository, vol. abs/1312.6170, December 2013

Source Model of TCP Traffic in LTE Networks

Paweł Foremski, Michał Gorawski and Krzysztof Grochla

Abstract We propose a generator that represents traffic transmitted in mobile wireless networks. It is based on measurements of IP flows in a real, large-scale LTE network. The proposed tool generates TCP flow sizes and durations. We verify the generator by comparison with the traces available in the literature, and we propose application in discrete event simulators.

1 Introduction

The performance evaluation of wireless networks requires a proper representation of the traffic transmitted within the network. The realistic assessment of evaluated network architecture or networking mechanisms must be done under conditions representative for the analyzed scenario. In particular, the traffic transmitted through the network must have similar statistical properties as the traffic imposed in future working conditions. Therefore, a good modeling of traffic source is crucial for the performance evaluation of any network protocol.

The network traffic modeling problem has been analyzed for many years. The large number of models representing popular protocols and applications were proposed, like HTTP [2], video streaming [14], or online gaming [1]. As most of the traffic in the Internet is transmitted via HTTP, the novel models better representing this protocol are still being developed—see, e.g., [19]. The first traffic models were based on Poisson distribution, but the authors of [18] shown that it does not represent correctly the traffic characteristics. The Internet traffic has self-similar characteristics, thus a large number of more sophisticated methods were used to correctly represent the

P. Foremski (✉) · M. Gorawski · K. Grochla
Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences, Gliwice, Poland
e-mail: kgrochla@iitis.pl

M. Gorawski
e-mail: mgorawski@iitis.pl

P. Foremski
e-mail: pjf@iitis.pl

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_14

observed statistical properties: on/off processes, Hidden Markov Models (HMM), ARIMA processes, Wavelets, diffusion approximation and multifractals [3, 6, 10].

The wireless network traffic is currently rapidly growing: according to Cisco Visual Networking Index, the volume of data transmitted by mobile devices increased by more than 80 % in 2013 [17]. Mobile users use different applications and browse the web slightly differently than the users of regular PCs: according to [16], the largest fraction of traffic to mobile devices is multimedia, and the HTTP objects transmitted to mobile devices are on average larger than transmitted to other devices. A considerable part of the traffic on mobile devices is generated by the synchronization of mobile applications with the servers, such as, e.g., social networks, mail, or calendar applications. This type of traffic is not present on stationary computers, thus dedicated models representing the properties of mobile device traffic are needed for proper performance evaluation of wireless networking protocols and algorithms.

There are a few traffic source models dedicated for wireless networks evaluation. LiTGen [20] allows to reproduce accurately the traffic burstiness and internal properties over a wide range of timescales, but is limited to mail and P2P applications. Authors of [15] use Hidden Markov Models to represent different QoS classes of the network traffic, but the model is based on a small traffic trace gathered in laboratory from a WiFi network, thus is not representative per large-scale mobile network traffic. In [4], the authors propose a theoretical model of eNodeB traffic that considers 6 different parameters—for example number of subscribers, data and voice activity during the busy hour, and the bandwidth required for bearer sessions—but the model provides only rough estimates on the aggregated traffic volume of a typical eNodeB. To the best of our knowledge, we were not able to find a traffic source model based on large-scale measurements in the 4th generation wireless network, i.e., Long-Term Evolution (LTE) networks.

2 Proposed Model

In this work, we propose a traffic generator representative for modern mobile devices traffic in LTE networks. It is based on measurements of traffic in a large-scale wireless network. The proposed tool generates TCP flow sizes and durations. We implement our model in the OMNeT++ environment and verify it by comparison to traces available in the literature. The model is based on a real-world research data [12], and thus generates traffic that is similar to transmissions in real mobile devices.

2.1 Network Traffic Data

Traffic traces are the key requirement for modeling network traffic. For example, popular methods apply HMM to model the interpacket time gaps and the packet lengths basing on observations of real traffic [6]. However, obtaining adequate samples of real-world LTE traffic is practically hard, so in our work we build on the data presented

Table 1 Characteristics of the real-world LTE dataset [12]

Capture start	Duration	Client IPs	Server IPs	Packets	Bytes	eNB count	Utilization	Flows (/5 min.)
October 12, 2012	240 h	379 K	719 K	3.8 G	2.9 T	22	29.5 Mbps	16.4 K

The column "Utilization" shows the average bandwidth used by all eNBs, while the "Flows" column shows the average number of traffic flows in 5-min time windows

in the recent paper by J. Huang et al., which gives an in-depth study of LTE performance [12]. We consider this work as representative and authoritative, as it presents the first study of a large real-world LTE packet trace. The traffic was collected from 22 eNBs in 2012 at a large metropolitan area in the U.S., and contains 2.9 TB of LTE traffic from over 300,000 users. The details on the dataset are presented in Table 1. For other papers that analyze wireless network traffic, see, e.g., [7, 8, 11, 13, 21]

The paper by Huang et al. characterizes several aspects related to the performance of LTE networks, but in our work we use the most fundamental data: the statistical characteristics of TCP session size and TCP session duration, given in Sect. 4.1 of [12]. As the authors point out, the TCP protocol dominates in LTE networks—carrying over 97 % of bytes and 95 % of flows—so we believe it is enough to model just the TCP protocol for a good approximation of the whole mixture of LTE traffic. Interestingly, for TCP, the pair of HTTP/HTTPS protocols is responsible for over 92 % of traffic flows.

In order to obtain the data required for our model, we digitized Figs. 2 and 3 from [12] and statistically processed the obtained data, as explained herein. After digitizing the Cumulative Distribution Functions (CDFs) of TCP session sizes and durations, we used the finite difference method to obtain their histograms and Probability Density Functions (PDFs), displayed in Figs. 1, 2, 3, and 4. Due to the range of the underlying data, density functions had to be presented in the logarithmic scale. This makes their interpretation harder, because integral of a PDF over its entire range must equal 1, but the corresponding histograms allow us to easily draw conclusions. The fit of the log-normal distribution is presented in Figs. 3 and 4 to roughly compare the data

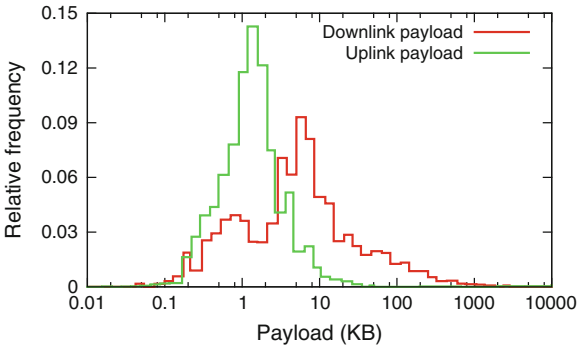


Fig. 1 Histograms of TCP flow sizes

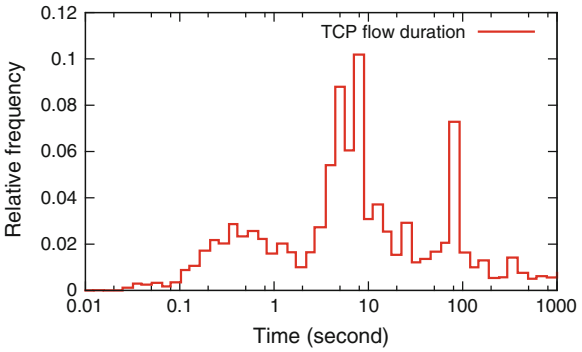


Fig. 2 Histogram of flow durations

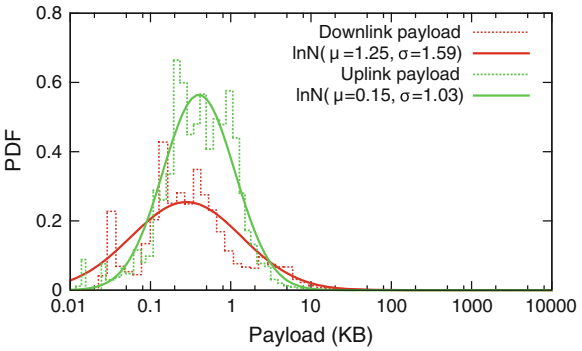


Fig. 3 Distribution densities of TCP flow sizes

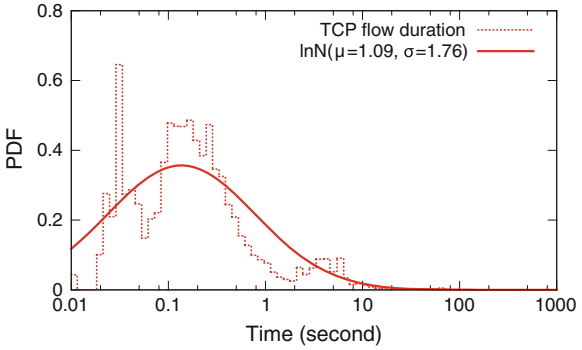


Fig. 4 Distribution density of flow durations

with a simple model. As visible in Fig. 4, the empirical distribution of flow duration cannot be explained using such a data model.

We chose to generate traffic flow data using inverse transform sampling on the digitized CDF plots. Each flow F is fully characterized with a tuple of $F = \langle S_u, S_d, D \rangle$,

where S_u and S_d describe the TCP payload size in the uplink and downlink direction (respectively), and D is the flow duration (i.e., the time from the first packet to the last packet). In order to linearly correlate $\log(S_d)$ and $\log(D)$ we used two correlated random variables X and Y for sampling the CDFs of S_d and D:

$$X \sim \mathscr{U}(0, 1), \tag{1}$$

$$Y = c \cdot X + (1 - c) \cdot Z, \tag{2}$$

$$Z \sim \mathscr{U}(0, 1), \tag{3}$$

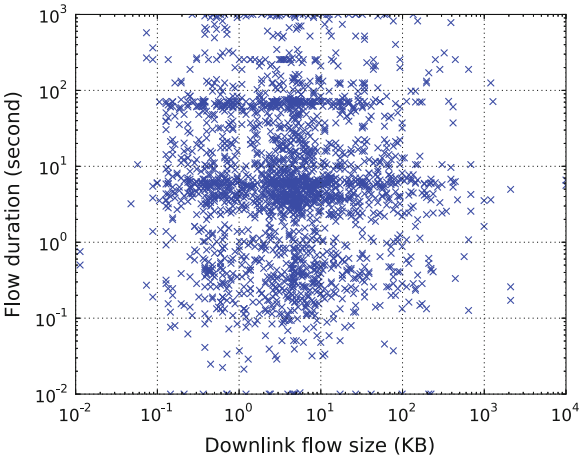


Fig. 5 Uncorrelated flow data

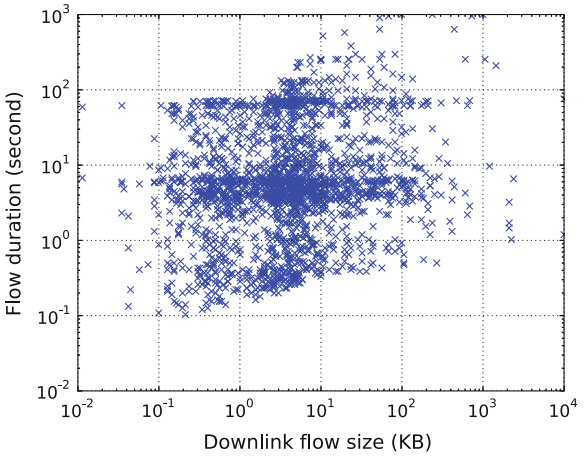


Fig. 6 Correlated flow data, $c = 0.16$

where $c \in [0, 1]$. The correlation of 0.196 reported in [12] was obtained for $c = 0.16$. See Figs. 5 and 6 for comparison between uncorrelated and correlated $\log(S_d)$ and $\log(D)$, for a sample of 2,500 pairs. For the uplink direction, we consider $\log(S_u)$ and $\log(D)$ as uncorrelated. Please refer to Chap. 7 and Example 2.1 in [5] for background on generating correlated random variates. We leave the task of describing F with a more sophisticated model for future work.

2.2 Network Traffic Model

To model the traffic from mobile devices, we used the OMNeT++/INET discrete event simulator, basing on the model of TCP connection described in [9]. To simulate uplink and downlink flows, we implemented two modules: a client (representing a mobile device) and a server (representing the Internet). Both modules were implemented as TCP applications (`TCPApps`) in the INET `StandardHost` module (Fig. 7). The `StandardHost` module enables simulating data flows in a TCP environment, along with a representation of all network layers. We assume no concurrent sessions, hence the communication is realized using a single pair of sockets. The sockets are closed after the session ends, and reopened for the next session: server socket is set to the `ListenOnce` state and binds to the client socket on connection request. Flow data is generated on the client side, where the correlated values described in Sect. 2.1 are

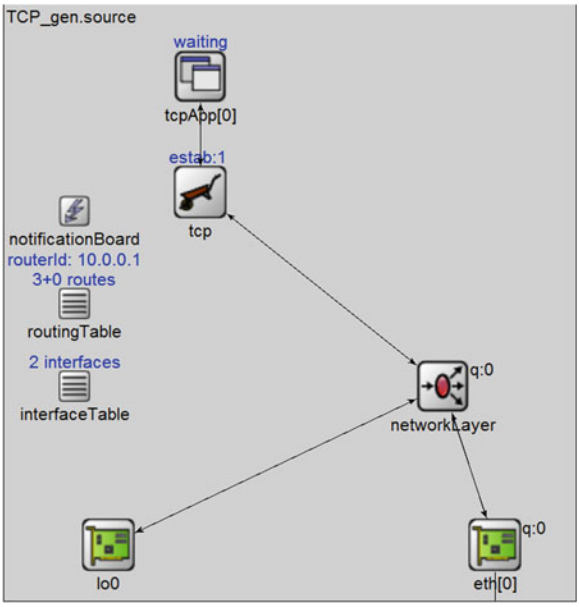


Fig. 7 The OMNeT++ implementation

drawn randomly. The packet transmission interval is set to 100 ms, and the packet size is constant for the whole session, so the total flow size matches the generated data.

The transmission is realized as follows. First, an uplink packet (named "Precast") is sent. It includes the information about expected response size and session duration. When the server receives the first packet it starts its own independent transmission with an adequate packet size. Transmission ends when both sides finish their transfers, and the session is immediately closed. In case of no uplink data, the client sends a single Precast packet and waits for the server to finish. For session times below 100 ms, the transmission is resolved by TCP modules; for session times between 100 and 200 ms, transmission is trimmed down to 100 ms. The interval between consecutive sessions is set by default to 1s, but it can be adjusted to any value or distribution, as needed. Note that this interval regulates the aggregated client bandwidth. Current implementation does not support simultaneous transmissions, which was left for future versions of our traffic generator.

3 Experimental Results

For experimental evaluation, we run our model implemented in OMNeT++ and collected statistics of 100,000 TCP sessions.

In Fig. 8, we compare the flow durations obtained from simulation with the expected flow data values, using a Quantile-Quantile plot (QQ-plot) of 1,000 samples. As visible, the distributions generally match each other, with an exception for the data in the range of 100–200 ms (on the OX-axis). This is expected, because our model presented in Sect. 2.2 divides data into 100 ms transfer windows, which limits

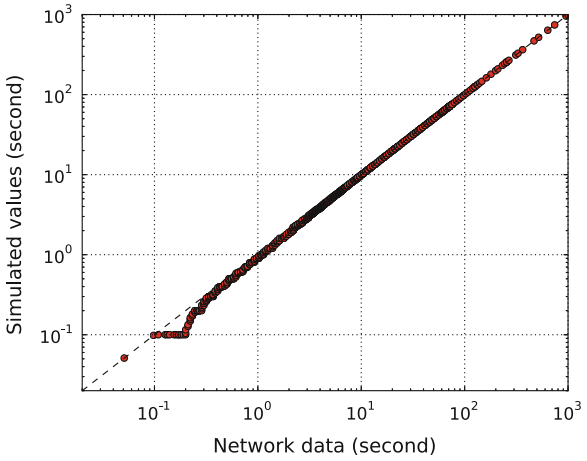


Fig. 8 QQ-plot of flow durations

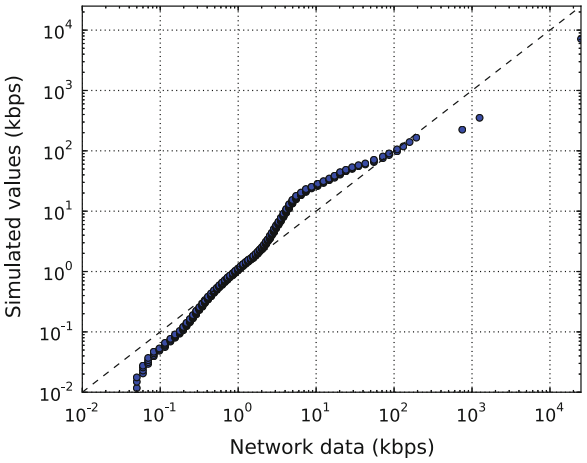


Fig. 9 Uplink flow rates (500 samples)

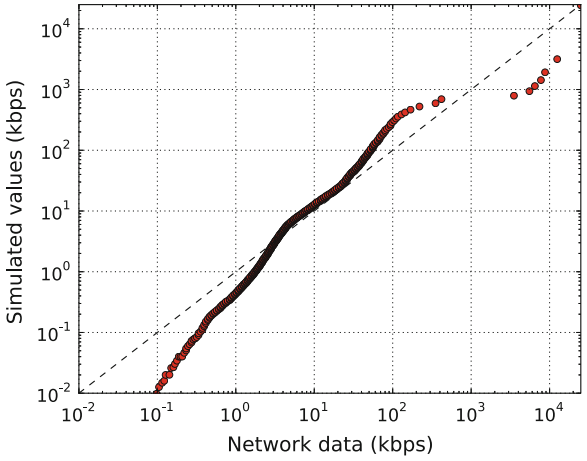


Fig. 10 Downlink flow rates (500 samples)

the granularity of flow durations observed in simulation. For larger flow durations this effect is negligible, hence not visible on the plot.

In Figs. 9, 10, 11, and 12 we compare the TCP flow rates obtained in OMNeT++ with the measured flow data shown in [12]. For both directions we see that the distributions are similar, with only minor deviations. Generally, slow flows (less than a few kilobits per second) and fast flows (more than a few megabits per second) tend to be transmitted with lower bit rates in the simulation environment. On the

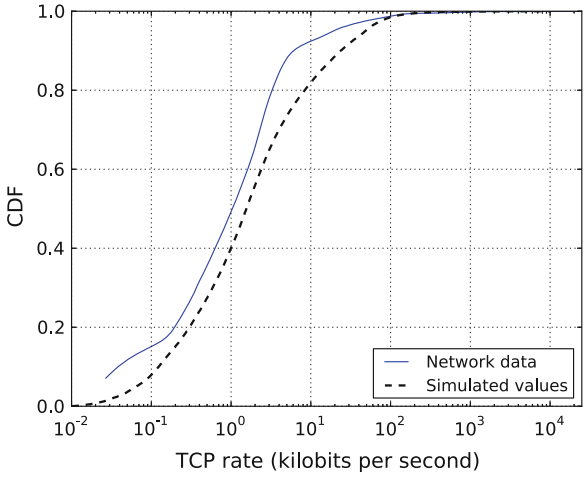


Fig. 11 CDF comparison of uplink flow rates

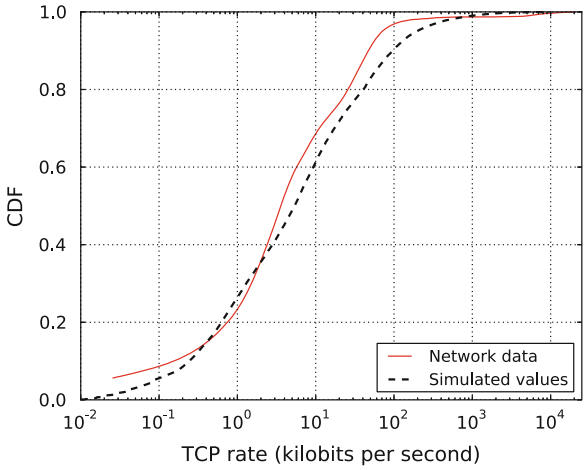


Fig. 12 CDF comparison of downlink flow rates

other hand, some flows in the middle are transmitted faster: for uplink the flows in the range of 5–100 kbps and for downlink the flows in the range of 50–1,000 kbps.

4 Conclusions

In this paper we proposed a model of TCP traffic that represents statistical properties of traffic in large-scale LTE networks, using the OMNeT++ discrete event simulation environment. The model is based on measurements of flow sizes and durations in a real network. The evaluation of the model confirms that it well matches with the traffic observed in the literature, providing similar distributions of transmission rates, for downlink and uplink directions.

The presented model can be used in various simulations as a source of TCP traffic generated by a mobile device working in an LTE network. We acknowledge that the model is based on the data collected in the U.S., whereas the behavior of the network users in other parts of the world may vary. However, as of this writing, there is no credible network data available, e.g., for European users, which may be an interesting avenue for future research.

The model can be downloaded at https://projekty.iitis.pl/rezultaty-badan-2-en.

Acknowledgments This work was funded by the Polish National Centre for Research and Development, under research grant nr LIDER/10/194/L-3/11/: project "Optimization and load balancing in next generation wireless networks", http://projekty.iitis.pl/zosb.

References

1. M.S. Borella, Source models of network game traffic. Comput. Commun. **23**(4), 403–410 (2000)
2. H. Choi, J. Limb, A Behavioral Model of Web Traffic, in *International Conference of Networking Protocol 99 (ICNP 99)*, pp. 327–334 (1999)
3. T. Czachórski, K. Grochla, T. Nycz, F. Pekergin, A diffusion approximation model for wireless networks based on ieee 802.11 standard. Comput. Commun. **33**, S86–S92 (2010)
4. D. Dababneh, M. St-Hilaire, C. Makaya, Traffic model for long term evolution networks, in *2013 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*, pp. 13–18. IEEE (2013)
5. L. Devroye, *Non-Uniform Random Variate Generation* (Springer, New York, 1986)
6. J. Domańska, A. Domański, T. Czachórski, Internet Traffic Source Based on Hidden Markov Model, in *Smart Spaces and Next Generation Wired/Wireless Networking, Lecture Notes in Computer Science*, ed. by S. Balandin, Y. Koucheryavy, H. Hu. vol. 6869 (Springer, Heidelberg, 2011), pp. 395–404
7. H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, D. Estrin, A first look at traffic on smartphones, in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 281–287. ACM (2010)
8. A. Gember, A. Anand, A. Akella, A comparative study of handheld and non-handheld traffic in campus Wi-Fi networks, in *Passive and Active Measurement*, pp. 173–183. Springer (2011)
9. K. Grochla, Simulation comparison of active queue management algorithms in tcp/ip networks. Telecommunication Systems 39(2), pp. 131–136 (2008) doi:10.1007/s11235-008-9117-x
10. H. Hlavacs, G. Kotsis, C. Steinkellner, Traffic source modeling. Tech. Heport Tr-99101 Inst. Appl. Comput. Sci. Inf. Syst. Univ. Vienna (1999)
11. J. Huang, F. Qian, A. Gerber, Z.M. Mao, S. Sen, O. Spatscheck, A close examination of performance and power characteristics of 4G LTE networks, in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pp. 225–238. ACM (2012)

12. J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z.M. Mao, S. Sen, O. Spatscheck, An in-depth study of LTE: Effect of network protocol and application behavior on performance, in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pp. 363–374. ACM (2013)
13. J. Huang, Q. Xu, B. Tiwana, Z.M. Mao, M. Zhang, P. Bahl, Anatomizing application performance differences on smartphones, in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pp. 165–178. ACM (2010)
14. M. Krunz, A. Makowski, A source model for VBR video traffic based on M/G/∞ input processes, in *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1441–1448. IEEE (1998)
15. S. Maheshwari, S. Mahapatra, C. Kumar, K. Vasu, A joint parametric prediction model for wireless internet traffic using Hidden Markov Model. Wireless Netw. **19**(6), 1171–1185 (2013) doi:10.1007/s11276-012-0525-1
16. G. Maier, F. Schneider, A. Feldmann, A First Look at Mobile Hand-Held Device Traffic, in *Passive and Active Measurement*, vol. 6032, Lecture Notes in Computer Science, ed. by A. Krishnamurthy, B. Plattner (Springer, Berlin, 2010), pp. 161–170
17. C.W. Paper, Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010–2015 (2011)
18. V. Paxson, S. Floyd, Wide area traffic: the failure of Poisson modeling. IEEE/ACM Trans. Networking (ToN) **3**(3), 226–244 (1995)
19. R. Pries, Z. Magyari, P. Tran-Gia, An HTTP web traffic model based on the top one million visited web pages. In: Next Generation Internet (NGI), 2012 8th EURO-NGI Conference on, pp. 133–139. IEEE (2012)
20. C. Rolland, J. Ridoux, B. Baynat, LiTGen, a Lightweight Traffic Generator: Application to P2P and Mail Wireless Traffic, in *Passive and Active Network Measurement*, vol. 4427, Lecture Notes in Computer Science, ed. by S. Uhlig, K. Papagiannaki, O. Bonaventure (Springer, Berlin Heidelberg, 2007), pp. 52–62
21. Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, S. Venkataraman, Identifying diverse usage behaviors of smartphone apps, in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pp. 329–344. ACM (2011)

A Few Investigations of Long-Range Dependence in Network Traffic

Joanna Domańska, Adam Domańska and Tadeusz Czachórski

Abstract The paper presents measurements and analysis of a LAN long-range dependence traffic collected in IITiS PAN (The Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences). Several methods of Hurst parameter estimation were used, the results obtained by the methods differ substantially. The analysis was made for the whole traffic and traffics generated by particular types of protocols. We seek for a dependence of Hurst parameter on a protocol type. Then, a MMPP (Markov-Modulated Poisson Process) model was applied to mimic the traces. It allows us to consider Markovian queueing models with long-range dependent and self-similar traffic, an important factor as we dispose an efficient software tool to solve numerically very large continuous-time Markov chains.

1 Introduction

During the last two decades, self-similarity and long-range dependence (LRD) became an important research domain [1–3]. Extensive measurements demonstrated the self-similarity and LRD of network traffic on several levels of communication protocols. Various studies made also evident that ignoring these phenomena in the analysis of computer networks leads to an underestimation of important performance measures as queue lengths at buffers and packet loss probability [4, 5]. Therefore, it is necessary to take into account this features in realistic models of traffic [6].

J. Domańska (✉) and T. Czachórski
Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
Baltycka 5, 44-100 Gliwice, Poland
e-mail: joanna@iitis.pl

T. Czachórski
e-mail: tadek@iitis.pl

A. Domański
Institute of Informatics, Silesian Technical University, Akademicka 16,
44-100 Gliwice, Poland
e-mail: adamd@polsl.pl

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_15

Section 2 contains a brief introduction to self-similarity and long-range dependance. Section 3 presents few methods of Hurst parameter estimation. Section 4 presents the Internet data collected in IITiS PAN and the analysis of LRD in these traces. Section 5 briefly explains how to fit two-state MMPP traffic model to second-order self-similar properties of observed data traces. Section 6 concludes this article and discusses future works.

2 Self-Similarity and Long-Range Dependence

Self-similarity is an often-observed natural phenomenon. The term was introduced by Mandelbrot [7] for explaining water level pattern of river Nile observed by Hurst. Let $X(t)$ be a stochastic process representing increment process (e.g., in bytes/second). In this case, X takes a form of a discrete time series $\{X_t\}$, where $t = 0, 1, \ldots, N$. The sequence $X^{(m)}(k)$ is obtained by averaging $X(t)$ over nonoverlapping blocks of length m:

$$X^{(m)}(k) = \frac{1}{m} \sum_{i=1}^{m} X((k-1)m + i), \qquad k = 1, 2, \ldots. \tag{1}$$

Let $Y(t)$ be a continuous-time process representing the traffic volume, i.e., $X(t) = Y(t) - Y(t-1)$. $Y(t)$ is exactly self-similar when it is equivalent, in the sense of finite dimensional distributions, to $a^{-H}Y(at)$, where $t > 0$, $a > 0$, and $0 < H < 1$ is the Hurst parameter. The process $Y(t)$ may be nonstationary [8]. The Hurst parameter H expresses the degree of the self-similarity [3].

Long-range dependence of data refers to temporal similarity present in the data. LRD is associated with stationary processes [8]. If a process $X(k)$ is second-order stationary with variance σ^2 and autocorrelation function $r(k)$, then it has LRD only if its autocorrelation function is nonsummable, $\sum_n r(n) = \infty$. That means that the process exhibits similar fluctuations over a wide range of timescales.

3 Hurst Parameter Estimators

Not all LRD processes mandatorily have a definable Hurst parameter, but the value of H between 0.5 and 1 is usually considered the standard measure of LRD [9]. The parameter can be estimated in a number of ways. The *R/S statistic, aggregated variance* and *periodogram* are well-known methods with a significant history of use, the *local Whittle's estimator* and *wavelet-based methods* are newer techniques and they perform relatively well.

The aggregate variance method [1, 10, 11] uses the plot of $\log[\mathrm{Var}(X^{(m)})]$ as defined in Eq. (1) versus $\log m$. The estimated value of Hurst parameter is obtained by fitting a simple least squares line through the resulting points in the plane. The asymptotic slope β between -1 and 0 suggests LRD and estimated Hurst parameter is given by $H = 1 - \frac{\beta}{2}$.

Another time-domain based technique of Hurst parameter estimation is called *R-S Plot* [7, 11]. The *R-S* method, one of the oldest techniques, is based on Central Limit Theorem. Let $R(n)$ be the range of the data aggregated over blocks of length n and $S^2(n)$ be the sample variance of data aggregated at the same scale. The rescaled range of X over a time interval n is defined as the ratio R/S:

$$\frac{R}{S}(n) = S^{-1}(n)\left[\max_{0\le t\le n}(X(t) - t\overline{X}(n)) - \min_{0\le t\le n}(X(t) - t\overline{X}(n))\right] \tag{2}$$

where $\overline{X}(n)$ is the sample mean over the time interval n and $S(n)$ is standard deviation. For LRD processes, the ratio has the following characteristic for large n:

$$\frac{R}{S} \sim \left(\frac{n}{2}\right)^H. \tag{3}$$

A log-log plot of $\frac{R}{S}(n)$ versus n should have a constant slope H as n becomes large.

The method using *Periodogram* in log-log scale [11, 12] is frequency domain method, the periodogram is defined by:

$$I_X(\omega) = \frac{1}{2\pi n}\left|\sum_{j=1}^{n} X_j e^{ij\omega}\right|^2 \tag{4}$$

A log-log plot $I_X(\omega_{n,k})$ versus $\omega_{n,k} = \frac{2\pi k}{n}$ should have a slope of 1–2H around $\omega = 0$.

Whittle's estimator is a semiparametric maxminimum likelihood estimator, which assumes a functional form to estimate the spectral density at frequencies near zero, [1, 13, 14]. To estimate Hurst parameter, one should minimize the function:

$$Q(H) = \sum_j \left[\log f_j(\omega_j) + \frac{\log I_X(\omega_j)}{f_j(\omega_j)}\right], \tag{5}$$

where $I_X(\omega)$ is the periodogram and $f(\omega) = c\omega^{2H-1}$.

The wavelet-based Hurst parameter estimators [1, 15] are based on the shape of the power spectral density function of the LRD process. Wavelets can be thought of as akin to Fourier series but using waveforms other than sine waves [11]. Wavelet analysis has been applied in Hurst parameter estimation due to its powerful properties.

4 Empirical Traffic Data Trace Analysis

Traffic measurements can be performed at various levels: byte, packet, flow, or session level. Source traffic at packet level has the following benefits as compared to the other level methods [16]:

- most of the network problems (loss, delay, jitter, etc.) occur at the packet level;
- packet-level approach is independent of protocols being used;
- traffic at the packet level remains observable even after encryption made by different protocols.

We used a trace of real Internet traffic collected from the network of IITiS institute serving < 50 academic users. This data set has been collected during the whole May 2012 on the Internet gateway [17]. The traffic approximately stands for a few dozen office users (researchers), mainly working Monday-Friday 8AM–4PM. IP packets were limited to 64 bytes—in most cases, they contain all headers plus a few bytes of the transport protocol payload. The local DNS traffic is not visible because of a specific setup of the network.

Captured packets are: packets leaving the local network and packets coming from the Internet that were not blocked by firewall. The collected informations were saved in tcpdump like format. The method of data collection allowed us the traffic classification and detection of the services which generated the most of traffic. The datasets contain different subsets of network protocols. The table shows the services and traffic generated by these services. The most of the traffic is associated with the TCP protocol, and normal HTTP generated over 17 % of traffic.

The estimators mentioned in Sect. 3 were used to evaluate the long-range dependence of the collected traces. Table 1 shows the results for one day traces (10/05—37 050 495 samples, 12/05—6 002 874 samples, 25/05—13 874 610 samples, 27/05—18 138 320, 31/05—36 135 490) obtained with the use of five estimators (250252 samples).

Table 2 presents the degree of LRD as a function of a type of protocol. This table gives the results for the dominant types of traffic for the most representative days. As can be seen, the security traffic is characterized by a smaller LRD. This observation

Table 1 Hurst parameter estimates for IITiS data traces

	31.05	27.05	25.05	12.05	10.05
Estimator	Hurst parameter				
R/S Method	0.721	0.74	0.655	0.498	0.763
Aggregate variance method	0.892	0.912	0.817	0.685	0.933
Periodogram method	0.678	0.781	0.715	0.613	0.84
Whittle method	0.683	0.714	0.599	0.473	0.761
Wavelet-based method	0.675 ± 0.14	0.681 ± 0.013	0.61 ± 0.027	$0/531 \pm 0.009$	0.71 ± 0.017

Table 2 Hurst parameter estimates for IITiS data traces

	31.05	27.05	25.05	12.05	10.05
Protocol	Hurst parameter—aggregate variance method				
HTTP	0.846	0.852	0.838	0.756	0.8
PPP	0.786	0.952	0.855	0.858	–
SSH	0.948	0.633	0.589	–	–
SSHv2	0.945	0.735	0.791	–	–
TCP	0.871	0.868	0.777	0.583	0.943
TLSv1	0.749	0.927	0.735	0.771	0.749

applies only to the selected type traffic analysis. The analysis of compound traffic (see Table 1) did not confirm the reletionship between the degree of LRD and the number of transmitted packets of a given type.

5 MMPP Model of LRD Traffic

Several models have been introduced to model self-similar and LRD processes in network traffic. The majority of them is based on non-Markovian approach. They use chaotic maps [18], α-stable distribution [19], fractional Autoregressive Integrated Moving Average (fARIMA) [20] and fractional Levy Motion [21] for modeling a network traffic. The advantage of these models is that they give a good description of the traffic with the use of few parameters. Their drawbacks consist in the fact that they do not allow the use of traditional and well-known queueing models and modeling techniques for computer networks performance analysis.

There are also Markov based models to generate a LRD traffic over a finite number of timescales [6, 9, 22–25]. This approach makes possible the adaptation of traditional Markovian queueing models to evaluate network performance. This section briefly describes the fitting of a superposition of two-state Markov-Modulated Poisson Process (MMPP) proposed in [26] to IITiS traffic data.

Two-state MMPP is also known as the Switched Poisson Process (SPP). The superposition of MMPP's is also an MMPP which is a special case of Markovian Arrival Process (MAP). A MAP is defined by two square matrices $\mathbf{D_0}$ and $\mathbf{D_1}$ such that $\mathbf{Q} = \mathbf{D_0} + \mathbf{D_1}$ is an irreducible infinitesimal generator for the continuous-time Markov chain (CTMC) underlying the process, and $D_0(i, j)$ (respectively $D_1(i, j)$) is the rate of hidden (respectively observable) transitions from state i to state j [27].

Following the model proposed in [26], a LRD process (used in our study) can be modeled as the superposition of d two-state MMPPs. The ith MMPP ($1 \leq i \leq d$) can be parameterized by two square matrices:

$$\mathbf{D_0^i} = \begin{bmatrix} -(c_{1i} + \lambda_{1i}) & c_{1i} \\ c_{2i} & -(c_{2i} + \lambda_{2i}) \end{bmatrix} \tag{6}$$

$$\mathbf{D_1}^i = \begin{bmatrix} \lambda_{1i} & 0 \\ 0 & \lambda_{2i} \end{bmatrix} \tag{7}$$

The element c_{1i} is the transition rate from state 1 to 2 of the ith MMPP and c_{2i} is the rate out of state 2– 1. λ_{1i} and λ_{2i} are the traffic rate when the ith MMPP is in state 1 and 2 respectively. The sum of $\mathbf{D_0}^i$ and $\mathbf{D_1}^i$ is an irreducible infinitesimal generator $\mathbf{Q}^i$ with the stationary probability vector:

$$\overrightarrow{\pi}_i = \left(\frac{c_{2i}}{c_{1i} + c_{2i}}, \frac{c_{1i}}{c_{1i} + c_{2i}} \right) \tag{8}$$

The superposition of these two-state MMPPs is a new MMPP with 2^d states and its parameter matrices $\mathbf{D_0}$ and $\mathbf{D_1}$ can be computed using the Kronecker sum of these d two-state MMPPs [28]:

$$(\mathbf{D}_0, \mathbf{D}_1) = \left(\oplus_{i=1}^{d} \mathbf{D_0}^i, \oplus_{i=1}^{d} \mathbf{D_1}^i \right) \tag{9}$$

There are two approaches for fitting the family of MAP to observed data: moment-based approach and likelihood-based approach [29]. The article [26] proposed a fitting method for a superposition of two-state MAPs based on Hurst parameter as well as the moments. All steps of this fitting procedure are precisely described in [30]. The fitting procedure requires the following input parameters:

λ^*—mean rate of the process to be modeled,
n—number of timescales,
d^*—number of active MMPP's,
$H = 1 - \beta/2$—the Hurst parameter,
ρ—lag 1 correlation.

Three of them: λ^*, ρ, and H should be estimated from the real data traces. Table 3 gives the parameters defining the model fitted using the set of descriptors obtained from IITiS trace. The superposition of four MMPP's is sufficient to model asymptotic second-order self-similarity of the counting process over five timescales.

Table 3 Obtained parameters of source fitted to the correlation structure of IITiS data (input parameters: $d = 4$, $n = 5$, $\lambda^* = 9.85$, $H = 0.71$ and $\rho = 0.021$)

	λ_i^{IPP}	c_{1i}	c_{2i}
IPP_1	1.124	4×10^{-1}	4×10^{-1}
IPP_2	0.4717	1.85×10^{-2}	1.85×10^{-2}
IPP_3	0.176	8.617×10^{-4}	8.617×10^{-4}
IPP_4	0.096	4×10^{-5}	4×10^{-5}
Poisson	$\lambda_p = 8.91551$		

6 Conclusions

Hurst parameter plays the key role in description of traffic LRD, which in turn influences the performances of a network. Its traditional estimators can be biased [15, 31]. Our study makes it evident that the values of H given by various approaches differ substantially. Our search for the dependence of H on the type of protocol does not give unambiguous results. Only Hurst parameter estimator based on wavelets can be treated as unbiased, efficent and robust [31]. Therefore, we used this method while fitting the parameters od MMPP models. Our future work will focus on the fitting to additional real traffic descriptors beyond second-order properties of the counting process.

Acknowledgments This work was supported by Polish project NCN nr 4796/B/T02/2011/40 *"Models for transmissions dynamics, congestion control and quality of service in Internet"* and the European Union from the European Social Fund (grant agreement number: UDA-POKL.04.01.01-00-106/09).

References

1. C. Park, F. Hernandez-Campos, L. Long, J. Marron, J. Park, V. Pipiras, F. Smith, R. Smith, M. Trovero, Z. Zhu, Long range dependence analysis of internet traffic. J. Appl. Stat. **38**(7), 1407–1433 (2011)
2. P. Loiseau, P. Gonçalves, G. Dewaele, P. Borgnat, P. Abry, P.V.B. Primet, Investigating self-similarity and heavy-tailed distributions on a large-scale experimental facility. IEEE/ACM Trans. Netw. **18**(4), 1261–1274 (2010)
3. Bhattacharjee, A., Nandi, S.: Statistical analysis of network traffic inter-arrival. In: Proceedings of the 12th International Conference on Advanced Communication Technology, ICACT'10, pp. 1052–1057. IEEE Press (2010)
4. Y. Kim, P. Min, On the prediction of average queueing delay with self-similar traffic. in: Proceedings of IEEE Globecom'03, vol. 5, pp. 2987–2991 (2003)
5. A. Gorrasi, R. Restino, Experimental comparison of some scheduling disciplines fed by self-similar traffic. Proc. IEEE Int. Conf. Comm. **1**, 163–167 (2003)
6. L. Muscariello, M. Mellia, M. Meo, M. Ajmone Marsan, R. Lo Cigno, Markov models of internet traffic and a new hierarchical mmpp model. Comput. Commun. **28**(16), 1835–1851 (2005)
7. B.B. Mandelbrot, J. Wallis, Computer experiments with fractional gaussian noises. Water Resour. Res. **5**, 228–267 (1969)
8. A. Nogueira, P. Salvador, R. Valadas, A. Pacheco, Markovian Modelling of Internet Traffic, *Network performance engineering* (Springer, Berlin, 2011), pp. 98–124
9. R.G. Clegg, Markov-modulated on/off processes for long-range dependent internet traffic. CoRR abs/cs/0610135 (2006)
10. J. Beran, Statistics for long-memory processes, *Monographs on statistics and applied probability* (Chapman & Hall, London, 1994)
11. R. Clegg, A practical guide to measuring the hurst parameter. 21st UK Performance Engineering Workshop, School of Computing Science Technical Report Series, CSTR-916, University of Newcastle pp. 43–55 (2006)
12. J. Geweke, S. Porter-Hudak, The estimation and application of long memory time series models. J. Time Ser. Anal. **4**, 221–238 (1983)

13. H. Kunsch, Statistical aspects of self-similar processes. in: *Proceedings of 1st World Congress Bernoulli Soc.*, pp. 67–74. VNU Science Press (1987)
14. P. Robinson, Gaussian semiparametric estimation of long range dependence. Ann. Stat. **23**, 1630–1661 (1995)
15. P. Abry, D. Veitch, Wavelet analysis of long-range-dependent traffic. IEEE Trans. Inf. Theor. **44**(1), 2–15 (1998)
16. S. Maheshwari, S. Mahapatra, C. Kumar, K. Vasu, A joint parametric prediction model for wireless internet traffic using hidden markov model. Wirel. Netw. **19**(6), 1171–1185 (2013)
17. P. Foremski, C. Callegari, M. Pagano, Waterfall: Rapid identification of ip flows using cascade classification, *Computer Networks*, Communications in Computer and Information Science (Springer, Switzerland, 2014), pp. 14–23
18. A. Erramilli, R. Singh, P. Pruthi, An application of deterministic chaotic maps to model packet traffic. Queueing Syst. **20**(1–2), 171–206 (1995)
19. J.R. Gallardo, D. Makrakis, L. Orozco-Barbosa, Use of α-stable self-similar stochastic processes for modeling traffic in broadband networks. Perform. Eval. **40**(1–3), 71–98 (2000)
20. F.C. Harmantzis, D. Hatzinakos, Heavy network traffic modeling and simulation using stable farima processes. in: 19th International Teletraffic Congress, pp. 300–303. Beijing, China (2005)
21. N. Laskin, I. Lambadatis, F. Harmantzis, M. Devetsikiotis, Fractional levy motion and its application to network traffic modeling. Comput. Netw. **40**(3), 363–375 (2002)
22. S. Robert, J.Y. Le Boudec, New models for pseudo self-similar traffic. Perform. Eval. **30**(1–2), 57–68 (1997)
23. J. Domańska, A. Domański, T. Czachórski, The drop-from-front strategy in aqm. Lect. Notes Comput. Sc. **4712**, 61–72 (2007)
24. J. Domańska, A. Domański, T. Czachórski, Internet traffic source based on hidden markov model. Lect. Notes Comput. Sc. **6869**, 395–404 (2011)
25. J. Domańska, D. Augustyn, A. Domański, The choice of optimal 3-rd order polynomial packet dropping function for nlred in the presence of self-similar traffic. Bull. Pol. Ac.: Tech. **60**(4), 779–786 (2012)
26. A.T. Andersen, B.F. Nielsen, A markovian approach for modeling packet traffic with long-range dependence. IEEE J. Sel. A. Commun. **16**(5), 719–732 (2006)
27. G. Casale, Building accurate workload models using markovian arrival processes. SIGMETRICS Perform. Eval. Rev. **39**(1), 357–358 (2011)
28. W. Fischer, K. Meier-Hellstern, The markov-modulated poisson process (mmpp) cookbook. Perform. Eval. **18**(2), 149–171 (1993)
29. H. Okamura, Y. Kamahara, T. Dohi, Estimating markov-modulated compound poisson processes. In: *Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools, ValueTools '07*, pp. 28:1–28:8 (2007)
30. J. Domańska, A. Domański, T. Czachórski, Modeling packet traffic with the use of superpositions of two-state mmpps, *Computer Network*, Communications in Computer and Information Science (Springer, Switzerland, 2014), pp. 24–36
31. C. Stolojescu, A. Isar, A comparison of some hurst parameter estimators. in: *Proceedings of Optim*, pp. 1152–1157 (2012)

Open Architecture for Quality of Service Monitoring at a National Research and Education Network

Alexandre Santos, M. João Nicolau, Bruno Dias and Pedro Queiros

Abstract The Portuguese National Research and Education Network (NREN) has a set of proprietary *appliances* for Quality of Service (QoS) monitoring probes within its backbone, very much dependant both on *hardware* and *software* details. Nowadays, several *open source* QoS monitoring systems and some Network Performance Measurement and Monitoring tools, developed both by GÉANT and Internet2 member institutions, are available. This work presents an open software architecture for generic hardware probes, based upon perfSONAR framework, for QoS monitoring and an associated management solution for software configuration and automatized distribution. This new architecture has been tested and deployed in a NREN backbone and QoS data has been gathered and integrated into the NREN's database. This paper discusses this open source solution for NREN QoS monitoring, the clock synchronisation issues and, finally, discusses the results obtained in a real testbed deployment in the Portuguese NREN backbone.

Keywords Network monitoring · Qos probes · Clock synchronisation

1 Introduction

The continuous increase in offered bandwidth enabled the emergence of new Internet based services, such as video-on-demand, and the new always-online approach, where best-effort Internet model is no longer appropriate and near real time and granted delivery is expected. So, generic Internet Service Providers (ISPs) and

A. Santos (✉) · M.J. Nicolau · P. Queiros · B. Dias
Centro ALGORITMI, University of Minho, Braga, Portugal
e-mail: alex@di.uminho.pt

A. Santos · B. Dias
Department of Informatics, University of Minho, Braga, Portugal
e-mail: bruno.dias@di.uminho.pt

M.J. Nicolau
Department of Information Systems, University of Minho, Braga, Portugal
e-mail: joao@dsi.uminho.pt

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_16

National Research and Education Networks (NRENs) show strong concern with monitoring their networks in order to early detect problems but also, and most specially, in order to monitor the service level of their network infrastructures. Apart from characteristics such as connectivity, available bandwith, packet loss, reordering or duplication and load balancing, very important timing parameters, such as delay and jitter, must be monitored.

This paper describes and justifies an open software architecture for generic hardware probes, based on the Multi-domain Network Performance Measurement and Monitoring, perfSONAR-PS [1], framework. This open architecture and its clock synchronisation issues are then analysed; the solution, its characteristics and results, are compared with those obtained with proprietary hardware *appliances* within the Portuguese NREN infrastructure, named RCTS. Section 2 presents related work on network monitoring and clock synchronisation, Sect. 3 presents open solutions for network probes and network probe management, Sect. 4 discusses the solution deployment in RCTS and analyses its operational results and, finally, Sect. 5 concludes.

2 Related Work

In general, monitoring may be accomplished by active or passive techniques. Active measurement implies the introduction of probing network traffic, thus enabling computing parameters such as end-to-end delay, delay variation between packets (*jitter*), packet loss or available bandwidth. Passive measurements, in turn, do not interfere with the network traffic: traffic is just (fully or partially) captured in specific network locations in order to be further processed (locally, or not) in order to extract monitoring data and statistics from it.

For a long time now, IETF has established the IPPM (IP Performance Metrics) Working Group that released documents specifying metrics and procedures to determine, among others, values for *one-way delay* or *one-way loss* mentioned in OWAMP [2]. Three OWAMP (One-Way Active Measurement Protocol) tools have been analysed: `owamp`, developed by Internet2, `QoSMet` [3] and `J-OWAMP` [4]. In [5] authors present a *hardware* solution that generates extremely precise clock synchronisation for OWAMP test packages. To compute metrics such as bandwidth, delay and packet loss, plenty of other tools exist, such as `iperf` , `nuttcp` and `thrulay` and the Internet2 tool `bwctl` can control all of these.

As for passive monitoring techniques, the most mentioned tools in the literature to make the capture and analysis of the traffic are `tcpdump`, `tcptrace` and `wireshark` but, in general, using generic hardware these are not able to cope with current real time line-speed analysis beyond 10 Gbps.

Several international projects have been carried out in order to study and develop QoS monitoring frameworks and tools. With larger scope, `perfSONAR` [1] was developed through an international collaboration between Internet2, ESnet (Energy Sciences Network), GÉANT (pan-European Research and Education Network) and RNP (Rede Nacional de Ensino e Pesquisa). `perfSONAR` is a service oriented

architecture that allows performance monitoring of networks, facilitating problem solving in end-to-end connections that traverse multiple networks. perfSONAR specifies a set of services and defines a protocol for communication between them; perfSONAR can be viewed as *middleware* allowing multiple implementations of different services to communicate with each other, thus widening the range of measurements that can be made, even between multi-domain users. Another project that should be mentioned is Archipelago [6] (or Ark), an infrastructure of active measurements led by CAIDA (Cooperative Association for Internet Data Analysis) which aims to reduce the effort required to develop and implement large-scale measurements, currently promoting the usage of inexpensive network measurement nodes, based on Raspberry Pi[1] devices.

Normally, one gets better accuracy when hardware timestamping methodologies are used. Nevertheless, methods based on *software* allow capturing traffic characteristics without additional costs, but errors may arise due to delays introduced by the Operating System (packets processing, buffers and interrupt management, for example) and different timestamping methods at OS kernel level, also influence its precision. Several studies on clock synchronisation have been performed, specially analysing synchronisation methods for packet switching networks. In this research area, three protocols are dominant: ntp, ptp and RADclock.

In the survey of network metrology platforms presented in [7], the authors analyse different existing methods to assess the Network Quality of Service. Designing an infrastructure to enable quality of service measurement in a network that is accurate, cheap and easier to implement remains a challenge.

3 Open Architecture for Network Monitoring

This section presents the proposed architecture for network monitoring in RCTS, discussing and justifying the main choices made. A set of laboratory tests conducted to determine the viability of using an Open Source solution and the results obtained are also presented and discussed.

A generic architecture is proposed, which can utilise both old *appliances* and generic hardware, integrating Open Source tools, namely perfSONAR, to establish a Probe-based QoS monitoring system.

The perSONAR framework consists of various tools (software implementations of various services), developed by various partners, and a protocol that assumes different types of services and defines a standard format and semantics whereby they communicate allowing different service implementations. This protocol is based on SOAP XML messages and was developed by the Open Grid Forum Network Measurement Working Group (OGF NM-WG). PerfSONAR services can run on

[1] Raspberry Pi, http://www.raspberrypi.org/, is a credit card sized ARM Linux computer that plugs into a TV and a keyboard.

multiple domains, using SOAP messages (carried in HTTP) to describe measurement data and to exchange information between services.

There are currently two major implementations of perSONAR: perfSONAR-MDM [8], developed by GÉANT and perfSONAR-PS [1], developed by Internet2 and ESnet. Both use the open protocol, sharing the same purposes: flexibility, scalability, openness and decentralisation. These two implementations differ in the development process, the life cycle of products, the interaction with users and the implementation model and distribution. perfSONAR-MDM has been developed as a multi-domain monitoring system, intended to provide a federated service, centrally monitored and coordinated, with full support from GÉANT. For perfSONAR-PS [1] transfer rate measurement is carried using `bwctl` and delay measurements are based on `owamp` [9]. The perfSONAR-PS has been selected on account of its good technical characteristics, good distributed support model that enables the proliferation of the number of nodes and also on account of its, ever increasing, large community of users.

The Network Time Protocol, NTP [10], was chosen as the time synchronisation protocol between probes due to its wide deployment and operating systems support, implementation flexibility and economic efficiency. Although an NTP distributed system implemented on top of a packet switch network will always yield out of synch probes in a non-determinist pattern, an adequately configured and maintained NTP system can ensure, at the present, a synchronisation methodology with a sufficient precision in regards to the scale of the precision of the expected probed values used for the computation of the referred QoS metrics in the context of the RCTS. OWAMP authors recommend that, at least, four `ntp` reference time servers should be used, although others [11] suggest that to maintain stability only one server, close to the probes and of low stratum, should be used. This one and only `ntp` server solution may result in smaller variations on probe time differences but it would also be a single point of failure of the synchronisation system. As such, it was decided that, from the four RCTS *stratum 1* `ntp` servers, two were to be used (one in Lisboa and the other in Porto) on the synchronisation system for the two probes, one probe in Lisboa and the other in Braga.

An open source solution derived from Red Hat Network (RHN) Satellite management for Linux systems, `spacewalk` [12], has been analysed and then selected in order to manage the whole set of perfSONAR probes, in a totally controlled and independent way. Spacewalk support for Linux OS derivative distributions like CentOS (apart from RHEL, Fedora, Scientific Linux) was also an important selective factor. This management component has been selected because it implements an open solution able to keep systems' inventory (both at hardware and software levels), manage software installation, maintenance and upgrade, enabling the establishment of custom software packages into manageable groups and to automate remote probe/system installation. Apart from these, `spacewalk` tool also allows the management of the individual configuration files for probes, the status analysis of each probe and even to control remote executions within any (even virtual) machine. This open solution includes a web UI, also allowing for command line and XML-RPC clients; at the data level, `spacewalk` is backed by either an Oracle or PostgreSQL database.

Fig. 1 Distribution of RCTS probes for QoS monitoring

4 Deployment and Results

Since 2006, RCTS has a set of probes, proprietary appliances that are no longer supported, which allows the measurement and control of the quality of service offered to RCTS users, including Universities, Research Laboratories and Polytechnics. The total number of probes comprises a grand total of 24 probes (see Fig. 1), having two

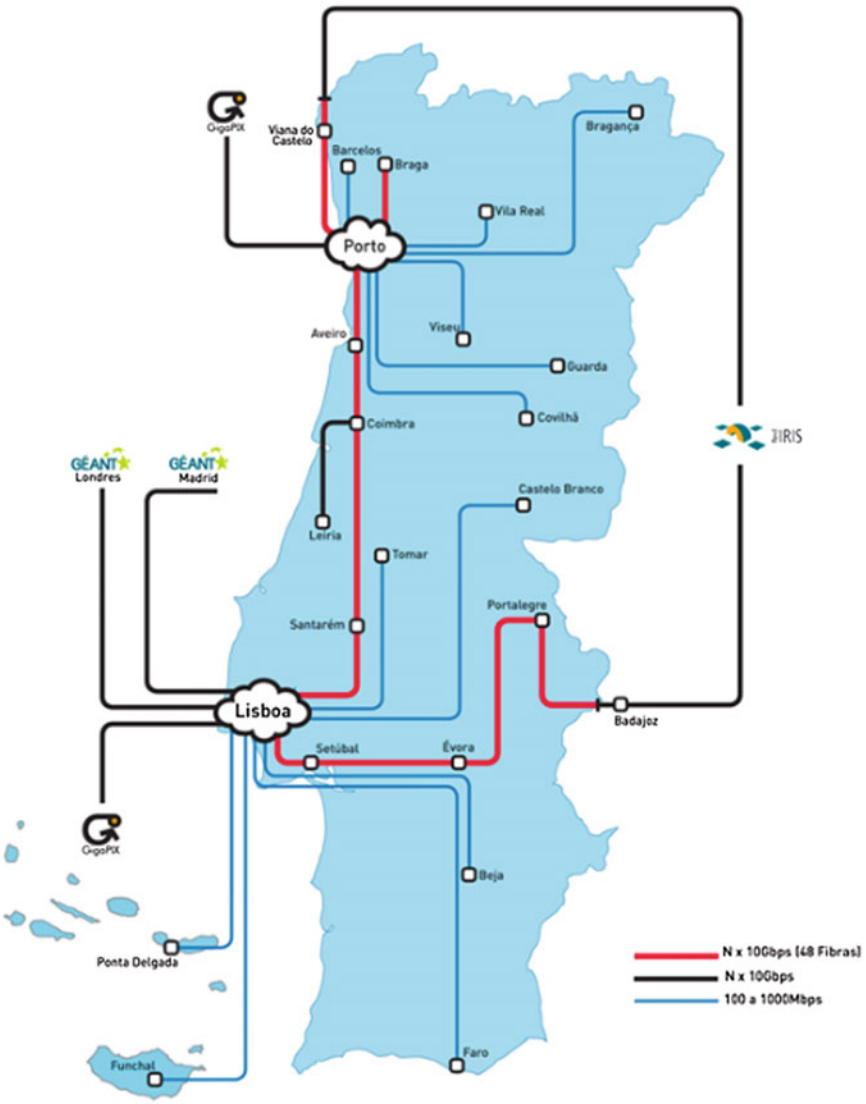


Fig. 2 The NREN RCTS Backbone

main probes, located in the central nodes of Lisboa and Porto, and several smaller probes localised at the interconnection points with academic institutions.

The Science, Technology and Society Network (RCTS) provides Portuguese researchers, professors, students and university staff with an advanced communication platform whose main backbone nodes and optical links are depicted in Fig. 2. For the deployment and for the real backbone test purposes here presented, it is important

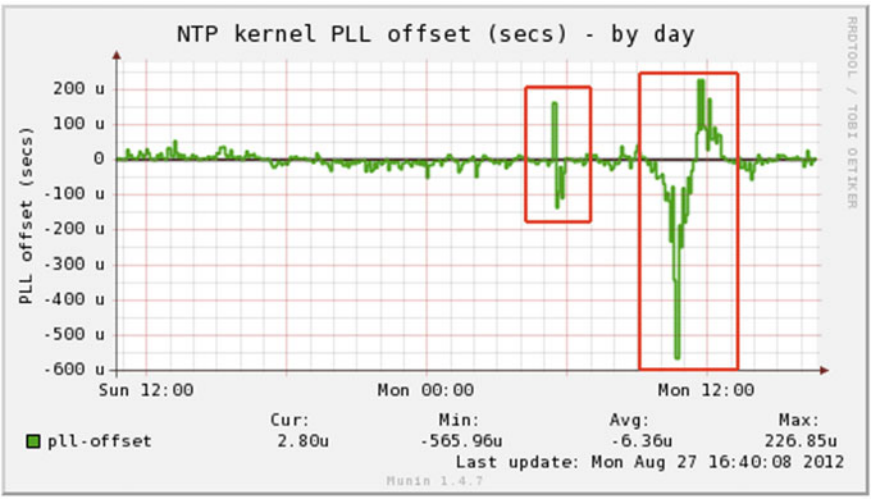


Fig. 3 NTP kernel PLL estimated error (temperature drop from 55 to 50 °C at CPU level)

to notice the three notorious testbed locations—Braga, Porto and Lisboa[2]—are interconnected by (at least) one lambda optical fibre link. Testbed deployment of perfSONAR based probes has been established via (level 2) connections directly into nodes in Braga and Lisboa, although the very first tests have been accomplished via a level 3 connection from Guimaraes UM Data Center to the RCTS node at Braga.

In order to evaluate our open source solution a very time sensitive parameter has been used, the measurement of one-way delay using the `owamp` component of perSONAR-PS v3.2.2. Indeed, `owamp` measurements rely on precise time synchronisation. For time synchronisation, `ntp` has been configured to use *stratum* 1 `ntp` servers (those `ntp` servers are directly connected (via optical fibre) to the RCTS NREN backbone nodes at Porto, Lisboa, Aveiro and Coimbra) bringing controlled topology [13] characteristics into the solution. By default, OWD measurements were taken from sending 10 twenty-byte long packets each (pseudo)session, each second interval. All performed tests showed up values for OWD between Guimaraes and Lisboa, both minimum and maximum values in each interval, extremely close to those obtained with the *hardware appliance*. Nevertheless, one could notice a few notorious results, namely some delay fluctuations and several delay spikes. A deeper analysis shows that there is a direct correlation between those fluctuations and the `ntp` kernel estimated error, as one can notice in Fig. 3 that those errors were directly connected with temperature fluctuations at the Guimaraes Data Center, due to the Data Center air conditioning daily cycles.

The probe has then been relocated to Braga, essentially to put it in a level 2 direct connection to the RCTS backbone but also to have a stable temperature of the probe at the Data Center, as Fig. 4 really shows (estimation error drops to a few μs).

[2] Geographic distances (aprox.): 50 Km for link Braga-Porto, 300 Km for link Porto-Lisboa.

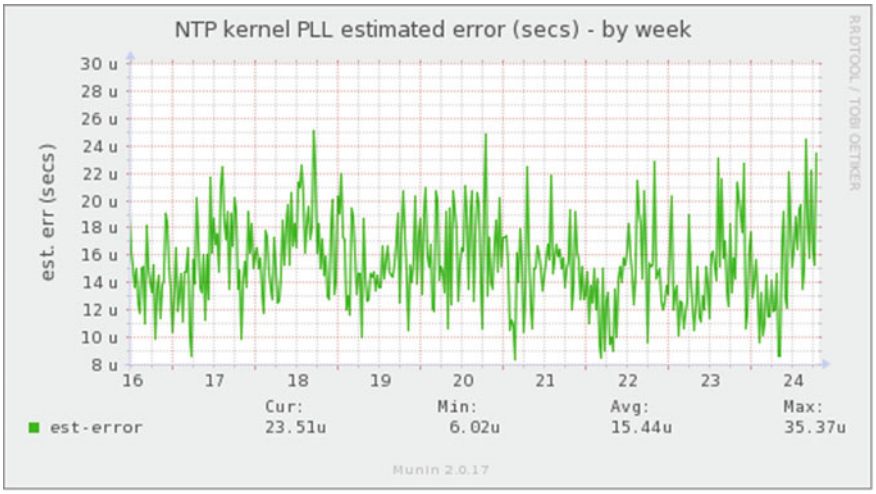


Fig. 4 NTP kernel PLL estimated error, RCTS probe node at Braga

The `ntp` itself estimates the OWD to the server as one half of the Round-Trip Time (RTT) and thus may introduce errors if the route is asymmetric, but it is not the case due to the characteristics of the NREN topology to those *stratum* 1 servers.

Also, one-way delay tests have now been configured to send a packet of 1,500 bytes every second, in order to match the requirements of the RCTS Service-Level Agreement (SLA). Figure 5 presents OWD measurements between RCTS nodes in Braga and Lisboa in this new setup. Results obtained with this new architecture are very similar to those in production (taken from the network of *hardware appliances* in RCTS).

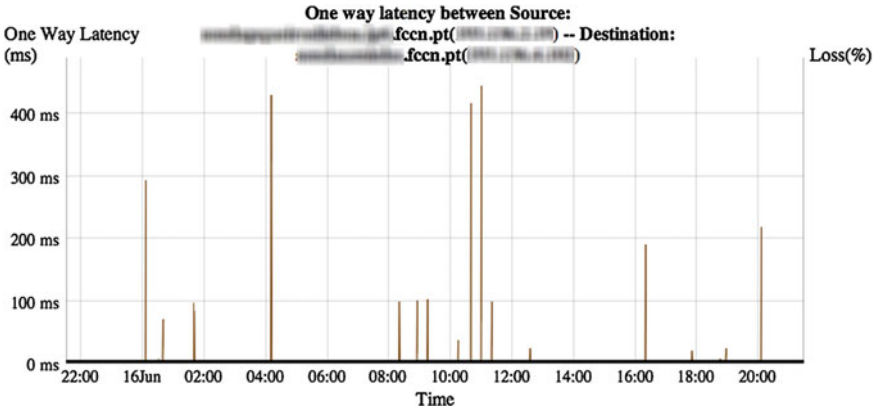


Fig. 5 One-way delay (packets w/1,500 byte): PerfSONAR probes Braga-Lisboa

The tests were performed both in IPv4 and native IPv6, as the NREN backbone is IPv6 capable and all the probes also have an IPv6 address. The results obtained for OWD measurements for IPv4 and native IPv6 are very analogous, although IPv6 delay values are, at the same time instant, generally almost 0.05 milliseconds higher than IPv4 delays, probably due to extra packet processing at kernel level.

Nevertheless, on account of large packet size and of delays in packet stack processing at kernel level, sporadic spikes appear in max OWDs, but those may be treated as *outliers* (still visible in Fig. 5) and may safely be statistically filtered out by using *percentile* 95 (in fact those spikes represent single occurrences in one minute measurements). This open software architecture has been validated against the production network, being that all the QoS parameters obtained with this final configuration have proved to be statistically equivalent to those that exist, at the same time, in the NREN production database.

Furthermore, as the QoS data extracted from each single probe had to be exported, consolidated and integrated into the production database of RCTS, a `php` script has been developed and installed at each perfSONAR probe, enabling the probe to export, every five minutes, the data collected into the NREN database repository. The data collected, taken from perfSONAR tables *NODES*, *DATA* and *DELAY*, is thus processed every five minutes to compute: `owd` (max, min and average) and its *percentile* 95, `jitter`, % of `lost packets`, # `sent packets` and # of `duplicates`, `ntp` estimated error and then sent to the DB repository.

Having all the open software solution implemented, tested and validated, one had to establish a central management scheme so that the NREN could be able to manage the whole set of perfSONAR probes, in a totally controlled and independent way. This management component has been implemented by means of `spacewalk`, an open source (GPLv2) Linux systems management solution. `spacewalk` is used in order to: keep systems' inventory (both at hardware and software levels); software installation, management and upgrade; manage groups of software packages; trigger and automate remote probe/system installation; manage probes' configuration files; control remote executions and manage virtual machines. Every probe system has to be registered into the `spacewalk` system (that collects its hardware characteristics) and, after a successful registration, the probe gets its `registration key` inheriting the associated `channel` properties. Every change made to a probe may be monitored and anytime reverted by means of the `snapshot` and `rollback` mechanisms. We have also used `kickstart` [14], a method for automatic remote installation of an operating system and all additional packages, through a file that contains the installation configuration (partitioning, packages to install, repositories to use, scripts to execute, etc.), thus freeing the administrator intervention during the installation process.

The framework we have established in order to measure and collect QoS parameters from the probes is easily extensible to other types of operational parameters and is also easily mapped into the backbone of any other organisations. Additionally, the centralised management framework we have established, using the `spacewalk` tool, proved to be suited for the task and easy to implement. Using `spacewalk` to create and manage local repositories of all packages needed by perfSONAR-PS

probes, also managing their specific configuration settings, is an important asset. This framework facilitates the deployment of this open source solution into different networks, managed by different NRENs.

5 Conclusions and Future Work

We have presented, tested and validated an architecture for the implementation of an *Open Source* solution, also using generic hardware and avoiding additional (internal or external) devices, to implement a network of QoS probes in RCTS NREN backbone. The whole open architecture, and its associated managing and reporting services, have been established in the RCTS *backbone* infrastructure. All the monitoring data sets obtained with this new open architecture have been validated against proprietary *appliance* data, with very satisfactory results.

The perfSONAR-PS has shown to be a good option for the collection of one-way delay (OWD) metrics through `owamp` tool, with the support of the `ntp` synchronisation protocol. No doubt that `ntp` may introduce some time measurement errors, being not the ideal clock synchronisation method for the whole network of QoS probes; indeed, it would be better to have all probes synchronised via an external (not network dependent, for instance via GPS or GSM synchronisation) source. The main problem with external clock synchronisation is neither a technological nor economic problem, it is rather a logistics problem on how to connect synchronisation antennas with the network equipment, normally in very different locations (e.g. antennas in the roof). Nevertheless, with a specific tunning of the `ntp` configurations and adequate topological locations of the *stratum* 1 ntp servers, and assuring that one can assure temperature stability, `ntp` has proved to be a valuable and standard tool for tenths of μs precisions (`ntp` should not be used if one aims to get precision at μs level).

The ability to access the database where the `owamp` tool stores the data computed from measurements allows an easy integration with other tools (e.g. the integration with the authoring tool for RCTS report generation was performed). However, as data goes to the database after being summarised, leads to accuracy loss and thereof to an inability to extract directly more useful information, such as delay variation. The existing dependence of `owamp` on `ntp` also influences the precision of measurements, it is desirable to drive `owamp` to support other time synchronisation methods.

As a centralised management tool, `spacewalk` proved to be a very suitable solution, not only for the support offered by the community but also for its (indirect) link to the CentOS operating system, used to support perfSONAR-PS probes. Nevertheless, the new open source solution that Red Hat's bringing for centralised management systems, project Katello [15], based on the concept of cloud, presents itself as the next logical evolution to analyse.

The open source architecture presented, whose full backbone operational deployment has been established (as time of writing), enables RCTS to perform the measurement of QoS parameters for SLA conformance. Furthermore, it offers the possibility

to extend the scope and range of current measurements, allowing a richer and more detailed analysis of the RCTS backbone, also fostering new interactions with other NREN implementations that are currently using perfSONAR.

Acknowledgments This work has been partially supported by FCT—Fundação para a Ciência e Tecnologia within the scope of the Project: PEst-OE/EEI/UI0319/2014. Authors would also like to thank the cooperation of FCCN, a branch of FCT managing the Portuguese NREN, for facilitating access to RCTS infrastructure.

References

1. The perfSONAR-PS Project, http://psps.perfsonar.net. Accessed 20 Dec 2013
2. S. Shalunov, B. Teitelbaum, A. Karp, J. Boote, M. Zekauskas, A One-way Active Measurement Protocol (OWAMP). RFC 4656 (Proposed Standard), Sept 2006
3. QoSMet - quality of service metrology, http://software.internet2.edu/owamp/. Accessed 20 Dec 2013
4. J-OWAMP: Java implementation of OWAMP, http://www.av.it.pt/jowamp/. Accessed 20 Dec 2013
5. Z. Shu, K. Kobayashi, Hots: An OWAMP-compliant hardware packet timestamp recorder, in Autonomic and Autonomous Systems and International Conference on Networking and Services, 2005. ICAS-ICNS 2005. Joint International Conference on, 2005
6. Archipelago measurement infrastructure, http://www.caida.org/projects/ark/. Accessed 20 Dec 2013
7. A.-C. Orgerie, P. Goncalves, M. Imbert, J. Ridoux, D. Veitch, Survey of network metrology platforms, in *12th International Symposium on Applications and the Internet (SAINT)*, 2012 IEEE/IPSJ, 2012, pp. 220–225
8. PERFormance Service Oriented Network monitoring ARchitecture for Multi-Domain Monitoring, https://forge.geant.net/forge/display/perfsonar/Home. Accessed 20 Dec 2013
9. OWAMP version 3.3 (an implementation of the one-way active measurement protocol), http://software.internet2.edu/owamp/. Accessed 20 Dec 2013
10. D. Mills, J. Martin, J. Burbank, W. Kasch, Network time protocol Version 4: protocol and algorithms specification. RFC 5905 (Proposed Standard), June 2010
11. W. John, S. Tafvelin, T. Olovsson, Passive internet measurement: overview and guidelines based on experiences. Comput. Commun. **33**(5), 533–550 (2010)
12. Spacewalk, http://spacewalk.redhat.com. Accessed 20 Dec 2013
13. S. Johannessen, Time synchronization in a local area network. IEEE Control Syst. **24**(2), 61–69 (2004)
14. Anaconda/Kickstart, http://fedoraproject.org/wiki/Anaconda/Kickstart. Accessed 20 Dec 2013
15. Katello, https://fedorahosted.org/katello/. Accessed 20 Dec 2013

Part IV
Security

Mitigating for Signalling Attacks in UMTS Networks

Mihajlo Pavloski and Erol Gelenbe

Abstract The setup of connections in mobile UMTS network will trigger signalling messages between the mobile and the core network. Malicious mobile phone or defective applications can therefore trigger *Signalling Attacks* which result in excessive wireless bandwidth utilisation and workload for the control plane and core network. We overview the cause of these attacks and identify the parameters which play a role.

Keywords Signalling attacks · Mobile networks · UMTS

1 Introduction

Smartphone and tablet allow users to access the Internet at any time and place [1], and this would function well if mobile phones had PC-like Internet connectivity. However, in the Universal Mobile Telecommunications System (UMTS) connections are dynamically created and teared-down for bandwidth allocation, generating traffic and signalling in the control plane of the network. Poorly designed or malicious applications can exploit this behaviour to create *Signalling Attacks* that can lead to documented network failures [2], by congesting the wireless bandwidth and signalling servers in the backbone, draining the user's battery and cause undesirable billing. We examine how network parameters can be set in order to lower the impact of signalling attacks, and investigate whether the network can maintain its stability under an attack by changing some specific state transition time constants, such as it's inactivity timers, or by adding delay in responding to bandwidth request messages.

We first review the related work in the field, then Sect. 2 briefly overviews the Radio Resource Control (RRC) mechanism in UMTS. In Sect. 3 we describe our

M. Pavloski (✉) · E. Gelenbe
Intelligent Systems and Networks Group, Department of Electrical and Electronics Engineering, Imperial College, London SW7 2AZ, UK
e-mail: m.pavloski13@imperial.ac.uk

E. Gelenbe
e-mail: e.gelenbe@imperial.ac.uk

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_17

model and its parameters, and in Sect. 4 we present and discuss the results obtained, and give directions for future work.

Security in general [3] has come to the forefront of much of the research in information technology in recent years, and cybersecurity in particular [4] is viewed as an integral part of security in general. Indeed it is impossible today to address physical security [5, 6] without including the impact of cybersecurity. In particular UMTS-based mobile network infrastructures which are universally available constitute an essential component of today's secure infrastructures.

The security of wireless networks has been of great interest in recent years [7], leading to many research projects in Europe and elsewhere [8]. In particular, signalling DoS attacks and their mitigation [9] has been a popular research topic in wireless and mobile communications. Publications in the field range from analytical algorithms, simulations using real-world data to complex systems for inspecting attacks on mobile networks, and the authors in [10] present an extensive survey of possible attacks in mobile networks.

A large Markov chain model is used in [11] for mathematical evaluation of signalling attacks' parameters, with the objective of identifying the system parameters which should be avoided, namely those that, from an attacker's perspective, produce the largest amount of damage through load in the network. The work in [12] regards the detection of traditional flooding-based DoS attacks as a change-point problem and applies the non-parametric CUSUM test for detection. Similarly, in [13] a CUSUM test in the early detection algorithm of low-rate, low-volume signaling attacks is suggested and simulations driven by real traces are used to demonstrate the impact of a signalling attack.

The work in [14] proposes a randomisation of the Radio Resource Management (RRM) and Mobility Management (MM) procedures to hide the parameters which are important to attackers. The analysis of signalling traffic in real-world UMTS network is presented in [15]. The paper shows a comparison of signalling traffic by different type of mobile applications and its influence on the RRC part of the network. It also explores some application and network layer solutions for controlling application signalling traffic. The authors in [16] inspect the influence of high signalling volumes in LTE networks on the energy consumption in mobile phones. Other modelling approaches of DoS attacks in 3G cellular networks are reviewed in [17].

2 UMTS Radio Resource Control

The management of communication resources in UMTS is regulated by the RRC mechanism. In general, there are two RRC connectivity modes: Idle and Connected. In Idle mode there aren't any radio resources used between the User Equipment (UE) and the Radio Network Controller (RNC). The few tasks a UE performs are related to neighbour cell monitoring, cell re-selection, paging and broadcast data reception. In this state, the UE consumes the least amount of energy. RRC's Connected mode is further divided in four states:

- CELL_DCH—a state where a dedicated connection exists in UL and DL direction. Radio resources are dedicated exclusively to the UE allowing it to send and receive data at higher speeds;
- CELL_FACH—there aren't any dedicated connections but data can be transferred via common channels. This state is suitable for transfer of small amount or bursty data. This state preserves the use of radio resources in the cell;
- CELL_PCH—similarly to Idle state the UE monitors only the paging and broadcast channels. The difference is that the logical RRC connection still exists;
- URA_PCH—a state similar to CELL_PCH where every cell change does not trigger a cell update procedure in order to decrease the signalling activity.

In UMTS the concept of connection is separated from the concept of Radio Bearer (RB). When an idle UE wants to make a data call it needs to establish a connection and obtain communication resources. The UE first initiates establishment of a RRC connection and then the network creates one or more RBs depending on the requested and available resources. The RB defines the properties of the connection depending on the requested QoS parameters. For instance, to transfer low-volume data the UE will obtain a common physical channel (CELL_FACH state) and a dedicated physical channel (CELL_DCH state) for a higher volume, delay-restricted data. The network then revokes allocated resources after an inactivity timeout t_L in CELL_FACH state or t_H in CELL_DCH state [18, 19].

The RRC mechanism, as described, is vulnerable to attacks triggering an excessive number of transitions between states. A single user request for connection/resources triggers multiple signalling messages that are transferred in the access and core part of the network. If requests are repeated regularly by many malicious UEs, the network will overload.

In particular, we can distinguish between two different types of signalling attacks:

- FACH attacks. A FACH attack occurs when the attacker makes a low-bandwidth request in repetitive intervals. This attack triggers signalling messages by transitioning between CELL_PCH and CELL_FACH states or between Idle and CELL_FACH states.
- DCH attacks. The attacker performs a DCH attack with repetitive high-bandwidth requests. This type of attack generates signalling traffic by alternating between CELL_DCH and CELL_FACH, CELL_PCH or Idle states.

The most common signalling attacks are CELL_PCH state triggered FACH attacks and CELL_FACH state triggered DCH attacks. Excessive signalling has negative influence also on the UEs because of high-power consumption.

3 System Model

The model used in this research is based on conventional stochastic modelling techniques [20] and focusses on a single user's RRC part of the UMTS system. It is described by the state diagram on Fig. 1. The figure depicts a model derived from

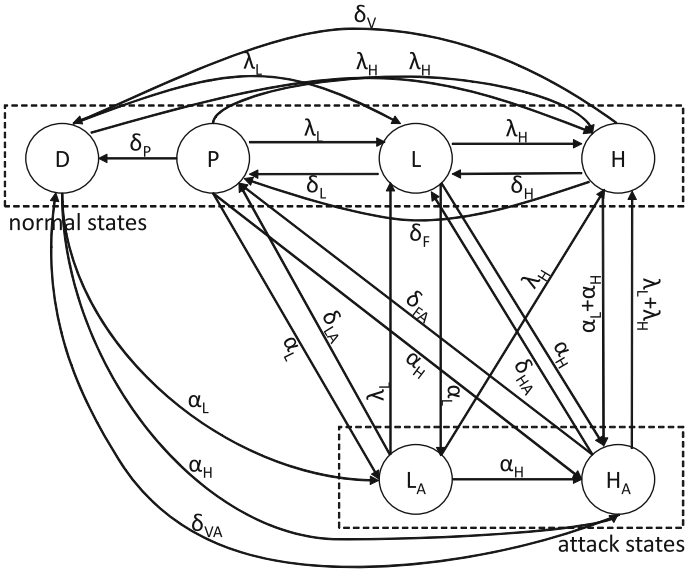


Fig. 1 RRC model of UMTS under signalling attack

the conventional UMTS model with added 'attack' states in the system. The idle state is represented by D—Dormant. CELL_PCH and URA_PCH are represented by a single P state. L (the low state) represents CELL_FACH and H (the high state) represents the CELL_DCH state. The corresponding states when attacks occur, or the attack states, are denoted with subscripts L_A and H_A for allocated FACH and DCH channels because of an attack.

Rates of promoting transitions are denoted with λ_i for normal and α_i for attacking bandwidth requests for state i. State demotion rates depend on the timeout intervals set at corresponding states and are given with δ_P, δ_L, δ_H, δ_{LA} and δ_{HA}. Transitions denoted by δ_F, δ_V, δ_{FA} and δ_{VA} represent the *fast dormancy* mechanisms which were introduced in later versions of UMTS standards. Two specific cases are included when low-bandwidth (FACH) requests are served in dedicated channel states, represented by the transitions from H to H_A and vice-versa.

4 Results and Discussion

First, we investigate the influence of the inactivity timers in FACH and DCH states, denoted with t_L and t_H respectively, on the security of the system. Three scenarios are inspected for both FACH and DCH types of attacks: t_L and t_H are changed together; the timer in DCH is fixed to 6 s and we change the timer in FACH; the timer in FACH is fixed to 4 s and we change the timer in DCH only. Then we denote with $t_{x\mathrm{L}}$ and $t_{x\mathrm{H}}$

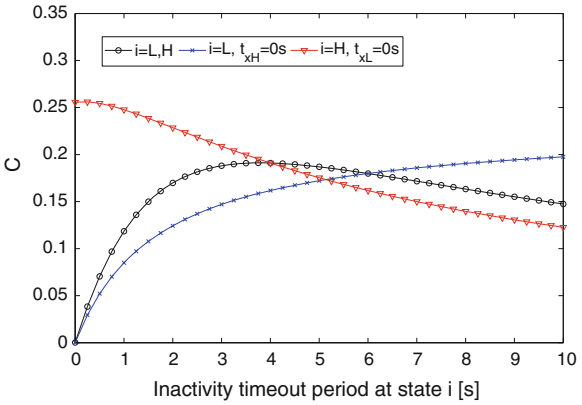


Fig. 2 The cost function C as a function of the inactivity timeout period

the inserted delay in transitions to corresponding states and examine three similar scenarios: inserting delay in both FACH and DCH requests; inserting delay only in FACH requests; and inserting delay only in DCH requests. In order to minimise the time spent in attack states and maximise the time spent in normal states, we define a cost function C as:

$$C = \frac{P_{\mathrm{LA}} + P_{\mathrm{HA}}}{P_{\mathrm{D}} + P_{\mathrm{P}} + P_{\mathrm{L}} + P_{\mathrm{H}}}. \tag{1}$$

We only present analytical results for FACH attacks to prevent repetition of similar results.

Figure 2 shows results from the three scenarios with regard to the inactivity timers in a system under FACH attack. For fixed $t_{\mathrm{L}} = 4\,\mathrm{s}$ the cost function decreases with the increase of t_{H} meaning that the longer the system stays in H state the lower the probability of FACH attack. For fixed $t_{\mathrm{H}} = 6\,\mathrm{s}$ the cost function increases with the increase of t_{L} meaning that the quicker the system returns to normal state, the lower probability of attack. The cost function for changing both t_{L} and t_{H} together rises to a certain point after which it starts to decline. Of course, the cost function has a minimum at $t_{\mathrm{L}} = t_{\mathrm{H}} = 0$ but selecting low values for the timeout periods would mean larger number of transitions (attacks) although the time spent in attacking states is minimised. Therefore, a better choice is selecting higher values for the two timers. In case of DCH attacks, changing the inactivity timeout in FACH state t_{L} does not influence the security of the system. Changing DCH, or FACH and DCH timers is similar to changing t_{L} and t_{H} timers together under FACH attack.

Figure 3 shows the influence of inserting delay in state promotion transitions in system under FACH attack. Setting $t_{x\mathrm{H}} = 0$ and increasing $t_{x\mathrm{L}}$ is a good choice for lowering the attack. Contrary to that, increasing the delay of DCH requests while an attack is ongoing on FACH state sharply increases the probability of attack states. Increasing the delay in both FACH and DCH requests at the same time improves the

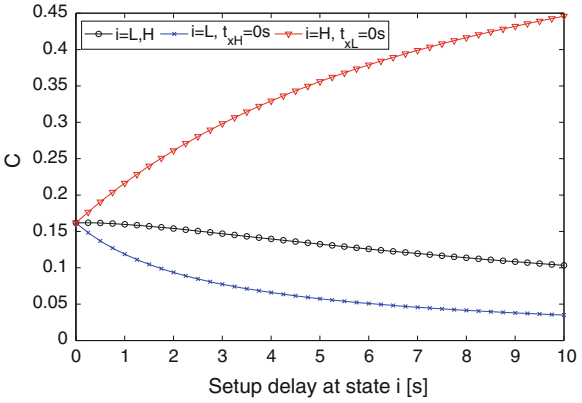


Fig. 3 The cost function C as a function of the setup delay at state i for FACH attacks

performance although not as much as increasing only $t_{x\mathrm{L}}$. Results for DCH attacks are analogous to the case of FACH attacks.

Future work may include analysis in a simulation environment as well as obtaining new mechanisms for mitigation of attacks. It would also be of interest to evaluate how such signalling attacks affect a realistic security setting when spectators at a sports or cultural venue are evacuated rapidly with the help of instructions given via smartphones [21–23] and a signalling attack launched by malicious individuals disrupt the evacuation.

References

1. P. Marck, IAB—Focus on mobile (2013), http://www.iab-community.be/iab-belgium-white paper-focus-on-mobile-jan-2013/
2. Rethink Wireless: DoCoMo demands Google's help with signalling storm (2012), http://www.rethink-wireless.com/2012/01/30/docomo-demands-googles-signalling-storm.htm
3. E. Gelenbe, F.J. Wu, Comput. Math. Appl. **64**(12), 3869 (2012)
4. E. Gelenbe, G. Görbil, D. Tzovaras, S. Liebergeld, D. Garcia, M. Baltatu, G.L. Lyberopoulos, in *ISCIS*. Lecture Notes in Electrical Engineering, vol. 264, ed. by E. Gelenbe, R. Lent (Springer, 2013), Lecture Notes in Electrical Engineering, vol. 264, pp. 369–378
5. Y. Cao, E. Gelenbe, in *Proceedings of the SPIE 3079, Detection and Remediation Technologies for Mines and Minelike Targets II*, p. 691, (1997). doi:10.1117/12.280898
6. E. Gelenbe, N. Schmajuk, J. Staddon, J. Reif, Robot. Auton. Syst. **22**(1), 23 (1997)
7. C.M. Yu, G.K. Ni, I.Y. Chen, E. Gelenbe, S.Y. Kuo, IEEE Trans. Inf. Forensics Secur. **9**(1), 109 (2014)
8. E. Gelenbe, G. Gorbil, D. Tzovaras, S. Liebergeld, D. Garcia, M. Baltatu, G. Lyberopoulos, in *Proceedings of the 2013 IEEE Global High Tech Congress on Electronics (GHTCE'13)* (2013)
9. E. Gelenbe, G. Loukas, Comput. Netw. **51**(5), 1299 (2007), http://dblp.uni-trier.de/db/journals/cn/cn51.html#GelenbeL07
10. G. Kambourakis, C. Kolias, S. Gritzalis, J.H. Park, Comput. Commun. **34**(3), 226 (2011). doi:10.1016/j.comcom.2010.02.010, http://dx.doi.org/10.1016/j.comcom.2010.02.010

11. O.H. Abdelrahman, E. Gelenbe, in *IEEE International Conference on Communications (ICC'14), Communication and Information Systems Security Symposium* (Sydney, Australia, 2014). Accepted for publication
12. H. Wang, D. Zhang, K. Shin, IEEE Trans. Dependable Secure Comput. **1**(4), 193 (2004). doi:10.1109/TDSC.2004.34
13. P. Lee, T. Bu, T. Woo, in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1289–1297, (2007). doi:10.1109/INFCOM.2007.153
14. Z. Wu, X. Zhou, F. Yang, in *2010 International Conference on Educational and Information Technology (ICEIT)*, vol. 1, pp. V1–504-V1-508, (2010). doi:10.1109/ICEIT.2010.5607650
15. Y. Choi, Ch. Yoon, Ys Kim, S. Heo, J. Silvester, IEEE Commun. Mag. **52**(1), 166 (2014). doi:10.1109/MCOM.2014.6710079
16. M. Gupta, S. Jha, A. Koc, R. Vannithamby, IEEE Commun. Mag. **51**(2), 90 (2013). doi:10.1109/MCOM.2013.6461191
17. F. Ricciato, A. Coluccia, A. DAlconzo, Comput. Commun. **33**(5), 551 (2010), http://dx.doi.org/10.1016/j.comcom.2009.11.015, http://www.sciencedirect.com/science/article/pii/S0140366409003168
18. 3GPP: UTRAN functions, examples on signaling procedures (Release 1999) (June 2002). TR 25.931 v3.7.0
19. J. Korhonen, *Introduction to 3G Mobile Communications* (Artech House Inc, Norwood, 2001)
20. E. Gelenbe, Acta Inf. **12**, 285 (1979)
21. A. Filippoupolitis, E. Gelenbe, in *2nd Conference on Human System Interactions, 2009. HSI'09*, pp. 323–330, (2009)
22. N. Dimakis, A. Filippoupolitis, E. Gelenbe, Comput. J. **53**(9), 1384 (2010)
23. G. Görbil, E. Gelenbe, in *PerCom Workshops*, pp. 540–546, (IEEE, 2013)

Cryptanalysis of a Cryptographic Algorithm that Utilizes Chaotic Neural Networks

Ke Qin and B.J. Oommen

Abstract This paper deals with the security and efficiency issues of a cryptographic algorithm which utilizes the principles of Chaotic Neural Network (CNN). The algorithm that we consider is the Delayed CNN-Based Encryption (DCBE), which is an encryption algorithm based on the Delayed CNN. Although the cryptographic algorithm has its own salient characteristics, our analysis show that, unfortunately, the DCBE is not secure since it is not capable of resisting known-plaintext, chosen-plaintext, and chosen-ciphertext attacks. Furthermore, unfortunately, the scheme is not efficient either, because of the large number of iteration steps involved in its implementation.

Keywords Chaos · Cryptograph · Chaotic neural network

1 Introduction

Over the last few decades, the phenomenon of chaos has been widely investigated and applied in a variety of domains including social networks, control systems, and prediction, etc. A chaotic system is characterized by salient phenomena

The work of this paper was supported by the National Natural Science Foundation of China (Grant No. 61300093) and Fundamental Research Funds for the Central Universities in China (Grant No. ZYGX2013J071).

B.J. Oommen is a *Chancellor's Professor* at Carleton University and a *Fellow of the IEEE* and a *Fellow of the IAPR*. He is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway.

K. Qin
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
e-mail: qinke@uestc.edu.cn

B.J. Oommen (✉)
School of Computer Science, Carleton University,
1125 Colonel By Dr., Ottawa, ON K1S 5B6, Canada
e-mail: oommen@scs.carleton.ca

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_18

such as its sensitivity to initial values, its pseudo-randomness and ergodicity, rendering it to be quite similar to a cryptographic system. The characteristics that render chaotic systems to be parallel or comparative to cryptographic algorithms are: Chaotic maps versus Encryption/Decryption algorithms, Iterations versus Rounds, Controlling parameters versus Keys, Sensitive to initial values versus Diffusion and Confusion, Pseudo-random, and ergodic.

As a result of the above observations, chaos has also been widely applied in the field of information security since Matthews proposed the first chaotic encryption algorithm [14] in 1984. Later, Baptista and Alvarez reported two cryptographic algorithms based on the phenomenon of chaotic searching in [1–3], respectively. While Erdmann et al. described a stream cipher based on the so-called Henon maps [4], Kanso and his coauthors illustrated a novel hash function [6] and showed how one could achieve digital image encryption based on chaotic maps [5]. Kocarev and his coauthors presented a public-key encryption [9] and random number generators [16] based on chaotic maps. A detailed list of articles that advocate the use of chaotic principles in cryptographic systems can also be found in [15, 19], and systematic reviews about chaos-based ciphers are found in [8, 10].

Now that chaotic *maps* have been proven to be useful in encryption, researchers have attempted to use Chaotic Neural Networks (CNNs), which are characterized by much more complicated dynamics than chaotic maps, to develop cryptosystems. The authors of [11, 12, 17] proposed different one-way hash functions based on different CNNs. Similarly, Yu and Cao proposed an encryption algorithm based on delayed CNNs [18]. Our present paper concerns some of these results.

Although the above-mentioned latter authors have affirmed that their schemes are secure and efficient, in this paper, we shall demonstrate that the security levels guaranteed by them are weak, and that they are inefficient. For example, most chaos-based ciphers require an excessive number of iterations, without which the ciphertexts are not sensitive to plaintexts. As opposed to these, traditional ciphers, e.g., the AES, only requires a 10-round calculation if one utilizes a key of 128-bits. Further, since chaotic equations are typically specified on the set of real numbers, the associated accuracy of implementing these schemes using digital computations is also problematic. Indeed, when we implement the associated computations numerically, we observe that some of the significant digits will be automatically truncated, and the consequence of this is that the original system which was chaotic within the domain of "real" numbers, is no longer chaotic [10]! Also, the improvement brought about by increasing the accuracy using higher-precision software entails a larger computational cost.

In this paper, we analyze an encryption method based on the so-called Delay Chaotic Neural Network (DCNN). However, we believe that our analysis is also valid for other CNN-based schemes.

2 The Delayed CNN-Based Cryptography

2.1 The Description the Delayed CNN-Based Cryptography

Delayed CNNs have been widely investigated in the past decades. The authors of [18] proposed a cryptographic system based on a special type of the delayed CNN. The model used in [18] is also a Hopfield-like NN which exhibits chaotic phenomenon and which obeys:

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = -c_i x_i(t) + \sum_{j=1}^{n} a_{ij} f(x_j(t)) + \sum_{j=1}^{n} b_{ij} f(x_j(t - \tau_{ij}(t))) + I_i(t), \quad (1)$$

where n denotes the number of units in the CNN, $x(t) = \{x_1(t), \ldots, x_n(t)\} \in R_n$ is the state vector associated with the neurons, $I = \{I_1, I_2, \ldots, I_n\} \in R_n$ is the external input vector, $f(x(t)) = \{f_1(x_1(t)), f_2(x_2(t)), \ldots, f_n(x_n(t))\} \in R_n$ corresponds to the activation functions of the neurons, $\tau(t) = \tau_{ij}(t)(i, j = 1, 2, \ldots, n)$ are the time delays. $C = \mathrm{diag}(c_1, c_2, \ldots, c_n)$ is a diagonal matrix, $A = (a_{ij})_{n\times n}$ and $B = (b_{ij})_{n\times n}$ are the connection weight matrix and the delayed connection weight matrix, respectively. The dynamics of Eq. (1) have been well studied and it is reported that it can exhibit rich chaotic phenomena [13, 20]. As demonstrated in [18, 20], if the parameters are:

$$A = \begin{pmatrix} 2.0 & -0.1 \\ -5.0 & 3.0 \end{pmatrix}, \quad B = \begin{pmatrix} -1.5 & -0.1 \\ -0.5 & -2.5 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

and if $f_i(x_i(t)) = \tanh(x_i(t))$, and $\tau(t) = 1 + 0.1\sin(t)$, $I = 0$, the trajectories of Eq. (1) are shown in Fig. 1.

The encryption and decryption schemes based on the above Eq. (1) proposed in [18] can be summarized as following:

1. Obtain the starting point x_0 from the last N_0 transient time iterations as $x_0 = x_1(N_0 h)$ where h is the discretized time step.
2. Divide the plaintext m into subsequences m_j of length l bytes, e.g., $l = 4$. That is, any message m can be digitized as:

$$m = \underbrace{p_0, p_1, \ldots, p_{l-1}}_{m_0} \underbrace{p_l, p_{l+1}, \ldots, p_{2l-1}}_{m_1} \cdots$$

where p_i is an 8-bit binary string.

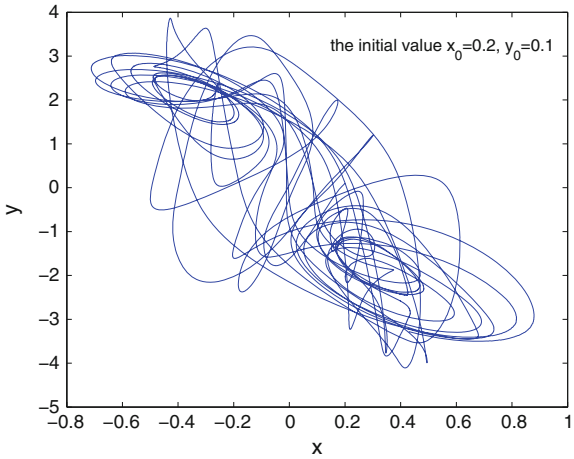


Fig. 1 The trajectories of Eq. (1). In this figure, the values of $x(t)$ and $y(t)$ are calculated by means of the fourth-order Runge-Kutta method. The time span is from 0 to 200 with a total of 30,000 steps

3. Combine four p_i to form a 32-bit binary block, implying that

$$P_j = p_j, p_{j+1}, p_{j+2}, p_{j+3}.$$

4. Iterate the initial value x_k for 38 times and to yield $x_{k+1}, x_{k+2}, \ldots, x_{k+38}$. Extract *one* bit from the 38 numbers and to obtain a 38-bit random binary sequence, $B_i = B_i^{k+1} B_i^{k+2} \cdots B_i^{k+38}$, where $B_i^k = b_i(x_k)$, is computed as per:

$$b_i(x_k) = \sum_{r=1}^{2^i-1} (-1)^{r-1} \Theta_{(e-d)(r/2^i)+d}(x_k), \tag{2}$$

and where e and d are the upper and lower bounds of x_k respectively.

$$\Theta_{\text{threshold}}(x_k) = \begin{cases} 0, & x_k < \text{threshold} \\ 1, & x_k \geq \text{threshold} \end{cases} \tag{3}$$

Denote $A_j = B_i^1 B_i^2 \ldots B_i^{32}$, $A_j^1 = B_i^{33} B_i^{34} \ldots B_i^{37}$, $A_j^2 = B_i^{38}$. Let D_j denote the decimal value of A_j^1.

5. Permute the message block P_j with a left cyclic shift D_j bits and the message block A_j with right cyclic shift D_j bits, to obtain P_j^* and A_j^*.
6. If $A_j^2 = 0$, the $x(t)$ is used for the successive block iteration illustrated in Step 4. Otherwise, $y(t)$ is used as the initial value of the next iteration.
7. Encrypt the message block P_j by XOR operations to yield:

$$C_j = P_j^* \oplus A_j^*. \tag{4}$$

where $\oplus$ is an XOR operation.

8. Reset the initial value by $x(0) = x(38+D_j)$ (or $x(0) = y(38+D_j)$, this depends on the value of A_j^2) and repeat the above steps till all blocks are encrypted.

As for the decryption, the steps are very similar to the encryption process except in the case of Step 7 where:

$$P_j^* = C_j \oplus A_j^*. \tag{5}$$

The plaintext P_j can be recovered by performing inverse permutations with right cyclic shifts of D_j bits.

2.2 The Analysis of the Delayed CNN-Based Cryptography

We now proceed to analyze the security and performance of the delayed CNN-based cryptography. We point out that this cryptography has several weaknesses:

1. **Non-randomness**:
   x and y are not uniformly distributed, which causes the "*random*" bits generated in Step 4 to be non-random. To illustrate this, we present the frequency statistics of the value of $x(t)$ and $y(t)$. The parameters used here are exactly the same as those used in Fig. 1. We categorize the combination of $x(t)$ and $y(t)$ into four classes:

   (a) $x \geq 0$ AND $y \geq 0$: 1,801
   (b) $x \geq 0$ AND $y < 0$: 15,618
   (c) $x < 0$ AND $y \geq 0$: 10,781
   (d) $x < 0$ AND $y < 0$: 1,800

   We can clearly see from the statistics that more than a half (52.06 %) of the $x(t)$ and $y(t)$ gather in the first quadrant, while only 48.94 % distribute in the other three quadrants. This phenomenon is confirmed from Fig. 1. Furthermore, as demonstrated in Step 4, we can normalize $x(t)$ and $y(t)$ into [0, 1] by:

$$g(x) = \frac{x-d}{e-d} = 0.b_1(x)b_2(x)\ldots b_i(x)\ldots b_n(x) \tag{6}$$

   where e and d are the upper and lower bounds of x respectively. We can thus generate the "random" binary bits according to $g(x)$. Indeed, the new counts are:

   (a) $b(x) = 0$ AND $b(y) = 0$: 2,769
   (b) $b(x) = 0$ AND $b(y) = 1$: 11,573
   (c) $b(x) = 1$ AND $b(y) = 0$: 14,379
   (d) $b(x) = 1$ AND $b(y) = 1$: 1,279

   Clearly, the bits generated by Eq. (3) are not "random".

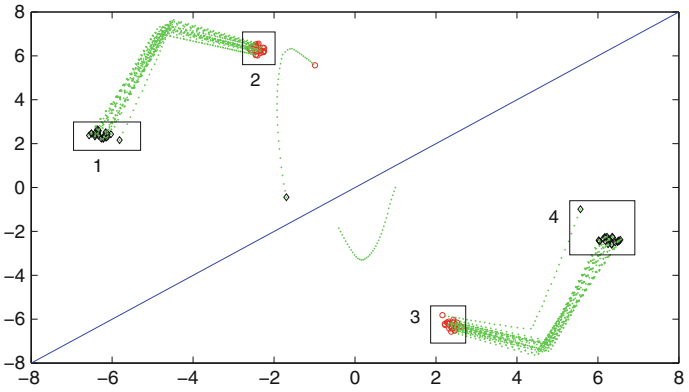


Fig. 2 The controlled trajectories of Eq. (1). For a better view, we have used a larger step 0.05 yielding a lesser number of points. Those points in *rectangle* 1 and 3, 2 and 4 are symmetric pairs along the axis given by the line $y = x$

2. **Trajectory behavior**:
The authors of [18] did not use the trajectories as shown in Fig. 1 directly. Instead, the random bits were generated according to the 38 successively iterations, as demonstrated in Step 4. We should thus carefully check the randomness of the corresponding sequences. According to Step 6 in Sect. 2.1, if $A_j^2 = 0$, $x(t)$ is used for the successive iteration, otherwise, it is $y(t)$. In this case, we swap the value of $x(t)$ and $y(t)$ every 38 iterations. As shown in Fig. 2 we can see that the value of $x(t)$ and $y(t)$ are very close during the 38 iterations, which means the random bits $B_i^1 B_i^2 \ldots B_i^{38}$ are probably the same. In spite of the above, the authors of [18] attempt to use this sequence to achieve the goals of "*diffusion*" and "*confusion*". It is well known that a sequence possessing poor randomness properties cannot be used in any cryptographic algorithm, because it would otherwise lead to a more predictable ciphertext. Consequently, we argue that this algorithm is not secure.

3. **Resistance to attacks**:
This cryptographic system cannot resist known-plaintext attack, chosen-plaintext attack, and chosen-ciphertext attack. To demonstrate this, assume that an attacker has some plaintext-ciphertext pairs (M_1, C_1), (M_2, C_2), and (M_3, C_3), where $\{M_i\}$ are the first 4 bytes of different plaintexts. If they are all encrypted by the same key, according to the algorithm, then A_j^*, D_j and some other intermediate iteration results should be the same. Thus:

$$C_1 = (M_1 \ll D_j) \oplus A_j^*$$
$$C_2 = (M_2 \ll D_j) \oplus A_j^*.$$

where $\ll$ denotes the cyclic left shift operation. Thus,

$$C_1 \oplus C_2 = (M_1 \ll D_j) \oplus (M_2 \ll D_j)$$
$$= (M_2 \oplus M_2) \ll D_j.$$

Since (M_1, C_1) and (M_2, C_2) are known, it is quite easy to find the value of D_j. After that, we can solve the equation $C_1 \oplus C_3 = (M_1 \oplus M_3) \ll D_j$ and thereafter determine M_3 successfully. Observe that during the whole process, we did not need any knowledge about the delayed CNN. The reason why we are able to proceed with such attacks is that the authors did not introduce the concept of the *Initial Vector* to the scheme.

4. **Efficiency**:
Although the authors of [18] claimed that the algorithm is efficient, this is not really the case. Actually, this conclusion is also true for many other cryptosystems such as those algorithms presented in [5, 7], which involve time delays in their equations. It is well known that the Runge-Kutta method is one of the best ways to solve differential equations where the initial values are provided. However, this method is still far too expensive when compared to traditional block ciphers such the DES or AES. Indeed, the computation of these traditional ciphers involves a finite field and only makes use of simple operations such as permutation. As opposed to this, solving differential equations involves the set of real numbers. For example, to encrypt a plaintext with size 1 M bytes, we have to divide the message into 1,024 × 1,024 /4 = 262,144 blocks, where each block is of length 4 bytes. According to Steps (1) and (4), at least $N_0 + 38$ iterations are involved to encrypt a single block. We have to thus do approximately $262{,}144 \times 100 = 26{,}214{,}400$ (here we assume that $N_0 = 62$) iterations to encrypt the whole file, which is, really, prohibitively large.

3 Conclusion

Chaotic Neural Networks have been widely used in various fields such as pattern recognition, dynamic associate memory, and optimization. Recently, cryptography based on chaos or CNNs has drawn great attention among researchers with chaos and cryptography interdisciplinary background. In this paper, we present a detailed analysis of a Delayed CNN-Based Encryption (DCBE). It analogous to common cryptographic algorithms, encrypts plaintext so that an eavesdropper will not be able to decrypt the message without the key. Although the authors have affirmed that this scheme is secure and efficient, our investigation proves that their claim is not valid. We have proven that the DCBE has been shown to not be secure since an attacker can partially recover the plaintext by using a known-plaintext attack, a chosen-plaintext attack, or chosen-ciphertext attack. We have also concluded that the scheme is not computationally efficient.

References

1. E. Alvarez, A. Fernandez, P. Garcia, J. Jimenez, A. Marcano, New approach to chaotic encryption. Phys. Lett. A **266**, 373–375 (1999)
2. G. Alvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of a chaotic encryption system. Phys. Lett. A **276**(1–4), 191–196 (2000)
3. M.S. Baptista, Cryptography with chaos. Phys. Lett. A **240**(1–2), 50–54 (1998)
4. D. Erdmann, S. Murphy, Henon stream cipher. Electron. Lett. **28**(9), 893–895 (1992)
5. A. Kanso, M. Ghebleh, A novel image encryption algorithm based on a 3d chaotic map. Commun. Nonlinear Sci. Numer. Simul. **17**(7), 2943–2959 (2012)
6. A. Kanso, H. Yahyaoui, M. Almulla, Keyed hash function based on a chaotic map. Inf. Sci. **186**(1), 249–264 (2012)
7. E. Klein, R. Mislovaty, I. Kanter, W. Kinzel, Public-channel cryptography using chaos synchronization. Phys. Rev. E **72**(1), 016,214 (2005)
8. L. Kocarev, Chaos-based cryptography: a brief overview. IEEE Trans. Circuits Syst. Mag. **1**(3), 6–21 (2001)
9. L. Kocarev, Z. Tasev, Public-key encryption based on chebyshev maps,in *Proceedings of the 2003 International Symposium on Circuits and Systems, Proceedings of the 2003 International Symposium on Circuits and Systems*, vol. 3, pp. 28–31 (2003)
10. S.J. Li, Analysis and design of digital chaotic ciphers, Ph.D. thesis, Xi'an Jiaotong University, China (2003)
11. Y.T. Li, S.J. Deng, D. Xiao, A novel hash algorithm construction based on chaotic neural network. Neural Comput. Appl. **20**(1), 133–141 (2011)
12. G.J. Liu, L. Shan, Y.W. Dai, J.S. Sun, Z.Q. Wang, One-way hash function based on chaotic neural network. Acta Physica Sinica **55**(11), 5688–5693 (2006)
13. H.T. Lu, Chaotic attractors in delayed neural networks. Phys. Lett. A **298**(2–3), 109–116 (2002)
14. R. Matthews, On the derivation of a chaotic encryption algorithm. Cryptologia **8**(1), 29–41 (1984)
15. K. Qin, M.T. Zhou, N.Q. Liu, A novel group key management based on jacobian elliptic chebyshev rational map. Lect. Notes Comput. Sci. **4672**, 287–295 (2007)
16. T. Stojanovski, L. Kocarev, Chaos-based random number generators—part i: Analysis. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **48**(3), 281–288 (2001)
17. Q.T. Yang, T.G. Gao, One-way hash function based on hyper-chaotic cellular neural network. Chin. Phys. B **17**(7), 2388–2393 (2008)
18. W.W. Yu, J.D. Cao, Cryptography based on delayed chaotic neural networks. Phys. Lett. A **356**(4–5), 333–338 (2006)
19. A.A. Zaher, A. Abu-Rezq, On the design of chaos-based secure communication systems. Commun. Nonlinear Sci. Numer. Simul. **16**(9), 3721–3737 (2011)
20. D. Zhang, J. Xu, Chaotic and hyperchaotic attractors in time-delayed neural networks. Complex Sci. **5**, 1193–1202 (2009)

DroidCollector: A Honeyclient for Collecting and Classifying Android Applications

Laurent Delosières and Antonio Sánchez

Abstract With the tremendous increase of Android malware, we need an automatic way of collecting Android applications and identifying the malware before they get installed on the end-user devices. In this paper, we propose a honeyclient for Android applications that will collect and classify Android applications. We first present an overview of the honeyclient. Then, we survey the different ways of infecting Android mobile devices which will shed the light on the honeyclient's design. Finally, we describe every component of the honeyclient, namely a crawler to build a list of suspicious URLs, a client to visit the suspicious URLs, extract Android applications and analyze them, and a malware detector to classify the collected Android applications. We use a light version of the Android browser to visit the suspicious URLs enabling us to scale the visits up and an Android emulator to analyze the Android applications. As for the malware detector, we use a combination of misuse and anomaly detector allowing us to detect already known malware and new variants.

Keywords Honeyclient · Android · Application · Collector

1 Introduction

With the tremendous increase of Android malware those last years [1], we need a way of collecting world widespread Android applications and identifying the malware before they get installed on the end-user devices.

Numerous Android-related papers have been published for the detection of Android malware such as [2, 3]. Some honeypots for Android have been proposed such as HoneypotLabSac [4] and honeyM [5] for passively collecting Android applications. To the best of our knowledge, we are the first to make the design of a honeyclient public for collecting Android applications by browsing the web.

L. Delosières (✉) · A. Sánchez
Hispasec Sistemas S.L., Málaga, Spain
e-mail: ldelosieres@hispasec.com

A. Sánchez
e-mail: asanchez@hispasec.com

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_19

This paper is proposing a honeyclient for collecting Android applications. As opposed to honeypots, honeyclients are actively searching for applications by visiting suspicious websites. We will first overview the honeyclient and its components. Then, we will survey the different ways of infecting a mobile device that will shed the light on the honeyclient's design. Finally, we will describe each of the honeyclient's components.

The contribution of this paper is the presentation of a honeyclient for collecting Android applications, i.e., (1) a crawler for crawling the web, (2) a client for visiting the web pages, collecting and analyzing Android applications, and (3) a malware detector for classifying the collected Android applications.

The paper is structured as follows: Sect 2 introduces the terms for understanding the article. Section 3 proposes an overview of the honeyclient. Section 4 surveys the different ways of infecting an Android mobile device that will shed the light on the honeyclient's design. Finally, Sects. 5, 6, and 7 will respectively describe a component of the honeyclient.

2 Background

GooglePlay [6], formerly called Android Market, is the official Android application store. It is accessible via the browser or via the Android application "Play Store". Alternative stores for Android applications or fake GooglePlays exist such as Amazon Appstore [7], GetJar [8], etc. For the rest of the article, we will interchange the terms Android application package file (APK) and Android applications.

We define a Points of Interest (PoI) as a resource that is commonly shared by a lot of people. A Point of Interest might be an online newspaper such as *bbc.com.uk* for instance. Since it is shared by a lot of people, it makes it ideal for an attacker to inject some malicious code and thus infect a lot of users.

A good malware detector presents a high detection rate and a very low false positive rate. It is worth mentioning that false positive rate refers to the rate of goodware that have been misclassified and treated as malware while detection rate corresponds to the rate of malware that have been correctly classified.

A C&C is a Command and Control server which is used by an attacker to control a malware installed on a remote device. It is used for sending commands to update a malware, steal information, etc.

VirusTotal is a free service for analyzing samples by 45 different antivirus engines. The service issues a report containing the malware name if so, the hash ID of the sample (MD5, SHA1, and SHA-256), the initial filename, the type of file (e.g., an image), etc. For some samples, the sample behavior is also inserted in the report. By behavior, we intend the actions of the malware on the system, such as the files that are read, written, the communications that are established, etc.

WebKit [9] is a framework to render HTML, CSS, and JavaScript. It provides an API that allows to interact with it for visiting webpages, rendering webpages, and downloading webpages and files. The WebKit core is composed of three main components: an HTML syntactic analyzer, a rendering engine, and a JavaScript interpreter.

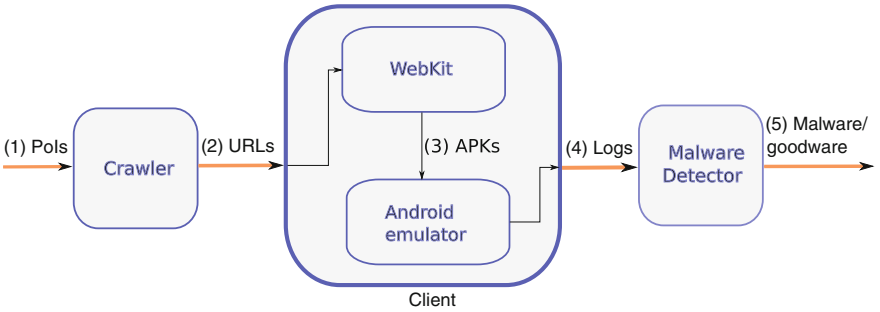


Fig. 1 Honeyclient

3 Honeyclient's Overview

The honeyclient is consisted of three main components, namely a crawler, a client, and a detector. In this section, we will show the place of every component in the honeyclient, and describe succinctly their role before describing them deeper in the next sections.

The whole honeyclient is depicted in Fig. 1. It first (1) receives a list of **PoIs** containing the list of web pages where it is very likely to be infected. This list is parsed by the crawler, described in Sect. 5, which crawls every **PoI** and their children, i.e., all the sublinks. All the extracted links compose the set **URLs** (2) which is sent to the client described in Sect. 6. The client is consisted of two modules, namely a WebKit that will visit every URL and an Android emulator. The former one will extract **APKs** (3), and send them to the Android emulator while the latter one will analyze the extracted **APKs** and will generate **logs** (4) out of the analyzes. Those **logs** will be sent to the malware detector described in Sect. 7 to classify the **APKs** as **malware or goodware** (5) from the **logs**.

4 Attack Vectors

In this section, we will enumerate the known attack vectors that will shed the light on the client's design.

In Android, as in other systems, there exists vulnerabilities. The most critical ones allow an attacker to take control over a system, extract sensitive information, or make a denial of service. In this study, we are only interested in the former one which allows an attacker to take the control over the device and install malicious software without the user's permission.

We have considered the vulnerabilities affecting the Android platform without the third-party software such as Adobe Reader, Java, etc. since they are not installed by default in most cases. In the future version of our honeyclient, we will also take

the third-party software into account. At the time of surveying the vulnerabilities (January 2013), there have been 29 exploits reported for the Android platform [10] whose critical vulnerabilities. However, the affected Android versions represent a minority of Google devices: only 1 % of the devices connected to any Google services. On the contrary, the versions between 2.3 and 4.1 represent an overwhelming majority of users with 89 % of the market. It is worth noting that we will take into account the future critical vulnerabilities in the next version of our honeyclient.

Since, the most used Android version, at the time of surveying the vulnerabilities, does not present any critical vulnerabilities publicly reported, we have been looking at other ways to infect mobiles. According to R.Unuchek [11] from Kaspersky Lab, there exits three main ways to distribute malware other than exploiting vulnerabilities:

- **Send bulk SMSs** (spam campaigns) that contain a malicious link. Upon clicking on the link, the user is asked to install a malicious application.
- **Fake Google Play**. Attackers create a page that looks like Google Play and distribute the fake Google Play's URL through internet. On the fake Google Play, e.g., blackmart, we only find free applications that are usually paying. Usually, those applications have been repackaged to integrate a malicious code.
- **Infect servers**. An attacker uses a persistent cross-scripting (XSS) vulnerability in a web page to inject a malicious JavaScript code. An example is provided in Listing 19.1. Upon loading a web page, the javascript code embedded on the web page is interpreted by the Android browser. It checks the browser's user agent and redirects the browser to a malicious application if the request comes from an Android browser, i.e., if the user agent contains the string "android".

```
window.onload = function () {
  if (navigation.userAgent.match ("/android/") {
    window.location = 'http://domain.com/fraudulent.apk'
 }
}
```

Listing 19.1 Javascript redirection

From the aforementioned information and the definition of the honeyclient, the fake Google Plays and the compromised servers turn out to be the best infection sources to collect Android malware. However, we will only take into account fake Google Plays that do not require any special Application Programming Interface (API) to collect Android applications, unlike Blackmart. Indeed, we want our crawler to mimic the users' behavior browsing the web.

5 Crawler

The crawler is responsible for crawling the web pages. We shall first see how we build the list of PoIs before showing the crawler's algorithm.

According to Symantec [12], there are PoIs' categories that are drawing the attention of attackers to infect devices and thus are very likely to contain malicious code. For instance, blogs turned out to be the first source of possible infection for Personal Computers (PCs). This is due to the easiness of compromising a blog and injecting some malicious code. Furthermore, blogs are very often visited and constitute a very attractive infection source for attackers. Blogs using frameworks, such as Joomla [13], WordPress [14], etc. are very commonly exploited because of the added plugins to the frameworks that present vulnerabilities.

Based on the Symantec's ranking, we have built a list of websites matching the most targeted sources which constitutes our list of PoI. We use the Algorithm 1 to browse each.... PoI, extract links, and send the extracted URLs to the Client.

The Crawler algorithm shown in Algorithm 1 takes as arguments a URL to visit and the depth of the visit. First, the URL is sent to the client via the function *SendURL-ToClient* that will visit it and extract eventual APKs. If the depth is different from 0, the links are extracted from the URL's web page which constitute the children. Each child is itself visited by calling the recurrent function *VISIT*. The algorithm ends up when all the children have been visited and their URL has been sent to the client, i.e., when the depth is equal to 0.

Algorithm 1 Crawler algorithm

```
procedure VISIT(URL, depth)
    SendURLToClient (URL)
    if depth == 0 then
        return
    else
        extractedURLs = ExtractURLs(URL)
        for each extractedURL in extractedURLs do
            VISIT(extractedURL, depth − 1)
        end for
    end if
end procedure
```

6 Client

The client consists of an Android emulator for running the Android applications, and making a dynamic and static analysis. For spaces reason, we invite the user to see our publication [15] describing the client for analyzing Android applications. However, for visiting the suspicious URLs, the client will use the browser WebKit enabling to

interpret Javascript code and download Android applications. As seen in Sect. 4, the attacker is very unlikely to use Android platform's vulnerabilities to infect mobile devices. As a matter of fact, we will use a light Android browser WebKit which allows us to visit URLs faster, making it more scalable, since we do not need to start and stop the Android emulators.

The information flow for WebKit is as follows: (1) it receives URLs from the Crawler, (2) visits the URLs and waits for about 10 s, the time to render the webpages and get redirected to possible other webpages, (3) iterates over all downloaded resources, and (4) sends any downloaded APK files to the Android emulator for its analysis. The rendering time was obtained through tests with fraudulent and legitimate URLs, and is the time required to obtain all resources of a web page taking into account possible webpage redirections.

In order to simulate a mobile device's browser, we have changed the *user-agent* of the WebKit by the Samsung Galaxy S2's one. The *user-agent* is sent in each HTTP request enabling a server to identify the client's browser. According to the *user-agent*, the server can propose different webpages. As seen in the Sect. 4, attackers use the *user-agent* to distinguish between mobile devices'browsers and computers' browsers. When a mobile device's browser is detected, a malicious Android APK is proposed instead of a webpage.

7 Malware Detector

The malware detector is consisted of an anomaly detector to detect new malware families and a misuse detector to detect already known malware. It will classify an Android application as malware if any of the detectors consider the application as malware and as goodware otherwise. We will see the dataset that will be used to train and test the malware detector before describing the anomaly and misuse detector.

7.1 Dataset

For training and testing the malware detector, we have used a dataset of 3,000 applications (about 1,500 malware and 1,500 goodware). All the samples have been collected from VirusTotal and third parties. We hypothesize that since those malware are one-year old, the signatures must have already been created. A sample was considered as malware if it was detected by at least 20 antivirus engines of VirusTotal and goodware if it was not detected by any of them. All the samples have been executed by the client which issued logs resulting from the static and dynamic analysis. Each log was transformed into a set of features.

7.2 Anomaly Detector

Before training and testing the anomaly detector, we have selected the most relevant features by using a feature selection algorithm. Selecting most relevant features enables to better model the goodware and malware and therefore get a higher detection rate and lower false positive rate. We have applied the feature selection algorithm proposed by Chein et al. [16]. It consists in building up feature vectors of different size. Each feature vector contains the most relevant features which are ranked by means of the F-Score metric. To evaluate the most relevant feature vector, the algorithm uses a Support Vector Machine (SVM) [17] with a gaussian radial basis function. After applying the algorithm on our dataset, the following features group has been selected and compose the feature vector:

- **SMS_functions**: 1 if the Android application contains functions to send SMS and 0 otherwise.
- **average_length_class_names**: the average length of class names.
- **average_entropy_class_names**: the average Shannon entropy of class names.
- **IMSI_functions**: 1 if the Android application contains functions to get the International mobile Subscriber Identity (IMSI) and 0 otherwise.

The feature **SMS_functions** characterizes Android malware that use premium SMS which represent an overwhelming majority of Android malware. The features **average_length_class_names** and **average_entropy_class_names** characterize Android malware that use obfuscation. Obfuscation is a technical employed for hardening reverse engineering and might be achieved by shortening class names and using a low Shannon entropy of class names. As for the last feature, it enables to get the phone ID.

We can notice that all the selected feature result from the static analysis. We were expecting it since the dormant code in Android malware is not as easy to execute even tough every Android application was instrumented by the client. Moreover, we suppose that during the analysis, the C&C were down preventing the malware to reveal its complete malicious behavior because the samples were one-year old.

We kept the same machine learning as Chein et al. [16] for our anomaly detector, that is a SVM with a gaussian radial basis function. The parameters of the SVM and the kernel function have been tuned according to the algorithm by Chein et al. [16]. For spaces reasons, we invite the reader to refer to their paper. The training set was composed of 300 malware and 300 goodware and the rest of the two sets was composing the testing set, i.e., over 80 % of the two sets (about 1,200 goodware and 1,200 malware). In order to know the unbiased performances of the anomaly detector, we have tested it on the same set randomized 200 times. On average, we get a detection rate of 90.5 % and false positive rate of 7.4 %. In the worst case, we get a detection rate of 86.2 % and a false positive rate of 10.1 %. In the best case, we get a detection rate of 93.6 % and a false positive rate of 5.2 %. It is worth noting that those performances do not represent the malware detector performances since we have omitted the misuse detector. With the misuse detector, we expect to

get a higher detection rate since it could detect the known malware that have not been classified as such by the anomaly detector.

7.3 Misuse Detector

Since, VirusTotal encompasses 45 different misuse detectors engines which combined together offer the best misuse detector of the world, we have elected it as our misuse detector.

In order to quantify the number of antiviruses that can potentially detect Android malware, we have empirically assessed it by taking randomly 1,000 Android malware from the malware dataset. Out of 1,000 Android malware taken from the dataset, only five antivirus engines were not able to detect any of the Android samples, namely nProtect [18], ByteHero [19], Malwarebytes [20], TheHacker [21], and SUPERAntiSpyware [22]. In other words, only 41 antiviruses were able to detect Android malware.

Like anomaly detectors, antiviruses suffer from false positive rate. In order to reduce the false positive rate, we consider an Android application as a malware if it has been detected by at least half of the antiviruses that were able to detect Android malware, i.e., 20 antiviruses.

8 Conclusion

This paper introduces a honeyclient for collecting and classifying Android applications as malware or goodware. We have first overviewed the components composing the honeyclient, namely a crawler which is responsible for crawling the web and getting suspicious URLs, a client that will visit the suspicious URLs, download potential Android applications and analyze Android applications, and a malware detector which will classify the collected Android applications as malware or goodware. We have designed the crawler to crawl Points-of-Interests that are resources shared by many people and where it is very likely to be infected such as Fake Google Plays. As for the client, it is based on a light version of the Android browser enabling us to have a scalable client for visiting suspicious URLs and on an Android emulator to run the collected Android applications. Finally, the malware detector is composed of both a misuse detector and an anomaly detector. The misuse detector is used for detecting known malware, based on the 45 antivirus engines of VirusTotal while the anomaly detector enables to detect new malware families based on a SVM machine learning. As future works, we will update the honeyclient by taking into account other threats, e.g., third-party software' vulnerabilities, etc.

Acknowledgments The work presented in this paper is funded by the European Commission FP7 collaborative research project NEMESYS (Enhanced Network Security for Seamless Service Provisioning in the Smart Mobile Ecosystem), no. 317888 within the Trustworthy ICT domain.

References

1. Android under siege: Popularity comes at a price (2012). Available: http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/reports/rpt-3q-2012-security-roundup-android-under-siege-popularity-comes-at-a-price.pdf
2. H. Gascon, F. Yamaguchi, D. Arp, and K. Rieck, Structural detection of android malware using embedded call graphs, in Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security, ser. AISec '13. New York, NY, USA: ACM, 2013, pp. 45–54. Available: http://doi.acm.org/10.1145/2517312.2517315
3. M. H. Daniel Arp, Michael Spreitzenbarth, K. R. Hugo Gascon, Drebin: Efficient and explainable detection of android malware in your pocket, in Proceedings of 17th Network and Distributed System Security Symposium (NDSS), 2014
4. V. B. de Oliveira, Honeypotlabsac: Um framework de honeypot virtual para o android, Master's thesis, PPGEE/UFMA, June 2012
5. T. J. O'Connor, B. Sangster, honeym: A framework for implementing virtual honeyclients for mobile devices, in Proceedings of the Third ACM Conference on Wireless Network Security, ser. WiSec '10. New York, NY, USA: ACM, 2010, pp. 129–138. Available: http://doi.acm.org/10.1145/1741866.1741888
6. Google play, 2013. Available: https://play.google.com/store
7. Amazon appstore for android, 2013. Available: http://www.amazon.com/mobile-apps/b?ie=UTF8&node=2350149011
8. Getjar, 2013. Available: http://www.getjar.com/
9. The webkit open source project, 2014. Available: http://www.webkit.org/
10. Cve details, 2013. Available: http://www.cvedetails.com/product/19997/Google-Android.html
11. Obad.a trojan now being distributed via mobile botnets, 2013. Available:http://www.securelist.com/en/blog/8131/Obad_a_Trojan_now_being_distributed_via_mobile_botnets
12. Symantec internet security threat report—2011, 2011. Available: http://www.symantec.com/threatreport/topic.jsp?id=threatreport&aid=malicious_code_trends_report
13. The platform millions of websites are buitl on, 2013. Available: http://www.joomla.org/
14. Wordpress.org, 2013. Available: http://wordpress.org/
15. L. Delosières, D. García, Infrastructure for detecting android malware, in Information Sciences and Systems 2013, ser. Lecture Notes in Electrical Engineering, E. Gelenbe, R. Lent, (eds.) Springer International Publishing, 2013, vol. 264, pp. 389–398. Available: http://dx.doi.org/10.1007/978-3-319-01604-7_38
16. Y.-W. Chen, C.-J. Lin, *Combining SVMS with Various Feature Selection Strategies, in Feature Extraction, Foundations and Applications* (Springer, Berlin, 2006), pp. 315–324
17. C. Cortes, V. Vapnik, Support-vector networks, Machine Learning, 20(3), pp. 273–297, 1995. Available: http://dx.doi.org/10.1007/BF00994018
18. nprotect, 2014. Available: http://nos.nprotect.com/
19. Bytehero, 2014. Available: www.bytehero.com/pages/english.html
20. Malwarebytes, 2014. Available: https://www.malwarebytes.org/
21. Hacksoft, 2014. Available: http://www.hacksoft.com.pe/
22. "Superantispyware", 2014. [Online]. Available: http://www.superantispyware.com/

Real Time Wireless Packet Monitoring with Raspberry Pi Sniffer

Yusuf Turk, Onur Demir and Sezer Gören

Abstract This paper proposes a real time wireless packet monitoring system using a Raspberry Pi. The system is a low cost alternative to commercial packet capture devices and analysis software. In our solution, captured packets from sniffer are sent to main server to gather statistics. Packets are analyzed and only the relevant data are stored in database. A notification server developed in Node.js provides communication between database and user interface developed with Django web framework. The performance of the proposed solution is successfully evaluated in an environment with multiple wireless networks. Results are presented.

1 Introduction

While the usage of Wireless Local Area Networks (WLAN) has been tremendously increasing, the cost of wireless networking devices has decreased. This enables to extend network in a more easier way. With the upcoming 802.11ac protocol supporting speeds competing with Gigabit Ethernet and increased usage of mobile devices have boosted the demand of WLANs. Access points are replacing the wired networks in homes and offices.

Considering the improvements in the networking technology, more and more devices are expected to have wireless networking capabilities. During the development of upcoming networked devices their networking capabilities and performance must be tested in an effective manner. Companies developing products with WLAN capabilities are in need of stable testing environments. The test environment may contain one or more sniffers to ensure the quality of transmission of 802.11 packets. Packet sniffers are used for comparing the expected network traffic with the

Y. Turk (✉) · O. Demir · S. Gören
Department of Computer Engineering, Yeditepe University, Istanbul, Turkey
e-mail: yturk@cse.yeditepe.edu.tr

O. Demir
e-mail: odemir@cse.yeditepe.edu.tr

S. Gören
e-mail: sgoren@cse.yeditepe.edu.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_20

captured traffic or analyzing the real time traffic. Because of the physical magnitude of WLANs, multiple packet capturing devices are distributed over the coverage area. Besides testing purposes, sniffers can also be used for security applications to detect intrusions by analyzing the traffic. Aside from that, both wired and wireless network monitoring, bandwidth utilization, and statistics gathering can be done with sniffers. Sniffers are useful because they allow several mechanisms to analyze the traffic thoroughly.

A study by Anh and Shorey [1] reviewed the current network sniffing tools that are used to obtain data by using the network interface card (NIC) of a PC. According to this study, not all the network sniffing tools support monitoring wireless traffic. Jipping and Holland [2] demonstrated that a network sniffer and packet classifier can be developed in a high level language. Another study done by Shum et al. [3] showed that the location of the source can be estimated if the physical location of the access point is known. In this work, a custom software on a router is used to gather network traffic. Location estimation is done by calculating the signal strength. Following study [4] showed that Received Signal Strength Indicator (RSSI) can be given as an input to estimate the location. Shum and Ng [5] showed that there is a correlation between RSSI and inter-device distance. Henderson et al. [6] logged terabytes of data with an access point sniffer. Then the collected data are used to gather statistics about the network usage in a campus. Although this study provides information about network usage patterns, security related problems are not covered. Boughaci et al. [7] used a sniffer module to detect intrusions. But using many agents in the system will slow down the system and timing is very important for intrusion detection. The attacks can also be from an internal source according to Henders et al. [8]. Data are read by a software tool running on NIC and written to a MySQL database.

There are numerous open-source or commercial capture software alternatives in the market. Tcpdump [9] and Wireshark [10] are the most popular open-source tools. Both of them offer detailed filtering options, but lack the visual analysis support. An expensive commercial alternative, OmniPeek [11], offer detailed visual analysis. Packet capture also done using OpenWRT firmware running on the supported hardware [12]. But many access points have lower CPU power and RAM size compared to Raspberry Pi. Access points with similar CPU values are more expensive than the Raspberry Pi. In a study by Polli et al. [13], Raspberry Pi with a USB wireless adapter is used to capture wireless packets. Although it is not used as a monitor mode sniffer, the study showed that the device is capable of capturing packets in a WLAN.

In this paper, we propose a low cost wireless packet sniffer using Raspberry Pi. Our solution is easily customizable and uses inexpensive off the shelf components. In addition, the monitoring software provides real time statistics of the current wireless network traffic and offers simpler user interface for analysis using a lightweight packet database. It also offers traffic frequency information about nearby access points.

The outline of the paper is as follows: Section 2 presents the components of the proposed system, Sect. 3 gives the performance results, and finally Sect. 4 concludes the paper including the future work.

2 Proposed Solution

The proposed solution consists of three functions: capturing, parsing, and monitoring. Figure 1 shows the components of the system. Capture operation is handled by a Raspberry Pi [14] with one USB Wireless NIC (WNIC) running a libpcap (packet capture library) [15] application. The output of this application is sent to the main server periodically. In this main server, a parser application extracts statistics and information from the packets. Parser application stores the summary of the packets in a database. For the monitoring part of the solution, a notification server is implemented to support communication between the database and the user interface.

The general overview and the details of the solution will be explained in the following subsections. First, packet capturing and wireless networking modes will be discussed. In Sect. 2.1, the details of Raspberry Pi is given and how the sniffing packets is achieved is explained. Real time parsing of the capture file .pcap using a Python program is described in Sect. 2.2. The section also covers retrieval and storage of the packet information in a database. Section 2.3 covers information about how real time monitoring interface is designed and implemented.

2.1 Sniffing on Raspberry Pi

Packet capturing is the process of grabbing a copy of a packet off the wired or wireless network before it is processed by the operating system. If the packet capture interface

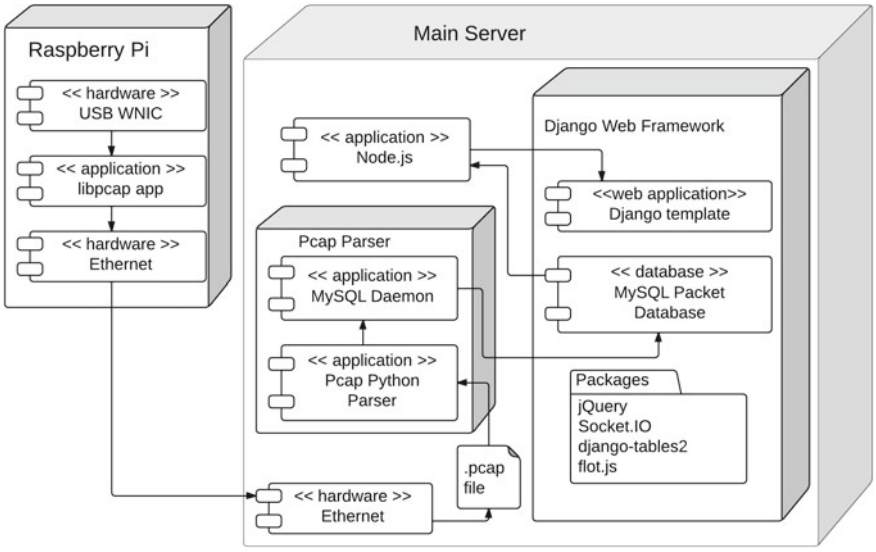


Fig. 1 Component diagram of the solution

is connected to a network, all frames are unencrypted and can be seen in the packet analyzing software. Otherwise, if the packets are captured in promiscuous or monitor mode, packets are still captured, but they are encrypted.

There are six operating modes of a 802.11 WNIC. These are master, managed, ad-hoc, mesh, repeater, and monitor modes [16]. Monitor mode is similar to promiscuous mode, but it is only applicable for wireless networks. Unlike promiscuous mode, devices do not have to be in a network. Monitor mode allows capturing of all the packets that can be seen by the WNIC. Monitor mode is dependent on the wireless adapter driver, firmware, and the chip-set features. Considering the limitations, not all the adapters support monitor mode. Sniffing of 802.11 packets can be done using a USB WNIC connected to a PC or using an access point. We have demonstrated the usage of both methods in our earlier work [17].

Raspberry Pi (RPi) is a low cost, small sized computer that can run many Linux distributions such as Debian, Fedora, and Arch Linux. With its Broadcom BCM2835 SoC including 700 MHz ARM processor and 512 MB RAM, RPi is more powerful and cheaper than most of the low cost off-the-shelf access points. The device does not have wireless interface onboard, but a USB WNIC can be easily attached to capture 802.11b\g\n packets. The mode of the WNIC is set to the monitor mode. A libpcap application in C is developed to capture packets. By adding the startup scripts, mode of the interface is set to monitor mode and the device starts to capture packets immediately. This makes the RPi a wireless packet sniffer. The channel that the device is listening on can be specified in the startup script or a random channel can be selected. RPi also supports multiple USB inputs, and a powered USB hub can be connected to the device. Multiple USB WNICs can be connected to this USB hub and each card may listen to another channel. The proposed work uses one USB WNIC connected to a RPi. Captured packets are forwarded to the specified Ethernet port of the device. Main server also listens to the same port and stores the raw data as soon as they are received. Port listening is done by using the netcat utility. Netcat is used to establish TCP and UDP connections and listen to a specified port.

2.2 Parsing Pcap Output

Packets received by the main server are appended to a file with .pcap extension. Pcap parser is developed in Python to parse the file in real time. Python is chosen because it is faster in reading files and easier to develop a parser with its standard library functions such as dictionaries. Main objective of the pcap parser program is to detect headers of each packet. Since all packets have the radiotap header by default, header detection is sufficient to identify the packets. Each time a header is detected in parser program, the subtype finding state starts and followed by parsing the access point information.

In the initialization state of the program log files, database tables, and variables are initialized. Since the objective is getting the real time monitoring, previous log files are truncated. MySQL database connection and a cursor is created at this state.

Capture file with the .pcap extension is read one byte at a time in the reading state. Each header block is sent to the header detection function and if a header is detected, state of the parser changes to header detected state. Detection of a header needs to be completed by finding the subtype of the frame. Subtype finding state can lead to SSID finding state if the frame is beacon, probe request, probe response, association, or reassociation frame. In SSID finding state, MAC addresses are retrieved. Subtype, SSID, and other information are logged in the logging state. Logging state is followed by the reading state. Program stays in the reading state until a header is detected.

Pcap parser writes statistical information directly to the database and information regarding the SSID and MAC addresses to a text file. This text file is simultaneously read by the MySQL daemon Python program. This is done to increase the parsing speed and avoid the large MySQL database queries in the main thread. It is intended to run MySQL queries at the background. MySQL queries can take long time when the data are significantly large. Therefore, only the data such as SSID and MAC address information of the frame are inserted into the database.

2.3 Real Time Monitoring

Node.js is a platform built on Google Chrome's Javascript runtime for building fast network applications [18]. Node.js is an appropriate solution for real-time applications. Since non-blocking models are used, thread does not have to wait for I/O. Although the platform can function as a web server, it can also be used as a server-side application. In this work, a Node.js application is developed to implement long polling functionality from Django web framework templates to MySQL database tables. Long polling is checking a data source for new data and instead of sending an immediate response, server waits until new data are available and then send the response. Long polling is an emulated version of push technology [19].

Node.js server and Django web framework do not have a default communication system. When new data pulled from the database server, it must be sent to the web application immediately to have real time monitoring. Socket.IO is used to provide communication between the two servers. Socket.IO is a Websocket framework that enables communication between WebSockets and real time monitoring. Websocket is a full duplex communication on TCP connection. Socket.IO can be used with Node.js, HTML, JQuery. The Node.js application creates a socket.IO interface on a specific port. Django application also listens from the same port. Node.js applications are event-driven. All listeners start at the same time and poll from the database tables at every given interval time. Server responds only when a new information is available. Socket.IO allows sending notification messages to one server to another.

Django is a Python web framework uses MVC (model—view—controller) pattern to create database powered web applications. Database tables are defined as models and implemented as Python classes. Web pages such as files with .html extension are called templates. Templates are redirected from the index page by using URL configuration file with regular expression support.

Monitoring of the captured data is supported with a user interface. First component displays the packet counters and current state of the parser. Second component displays the information regarding management frames such as SSID, dBm signal rate, data rate, and MAC addresses. Third component displays the averaged dBm signal of each SSID.

3 Performance and Results

Several test platforms are prepared for testing the system. First, a network with an access point with two Linux based computers run the iperf application to transfer UDP packets. While running the iperf client and server commands, Raspberry Pi sniffer captured the transferred packets without connecting to the network. An another access point, a computer and a mobile phone are also connected to a different network in the same test environment. In the second network, a video stream setup is prepared to provide constant streaming of videos from computer to mobile phone. Sniffer captured packets without connecting to any network. Figure 2 shows the initial test setup.

Main objective of the parser application is to generate statistics from the captured packets. Packets are appended to a file and it is read by the parser simultaneously. Program waits until new packets are available and runs in an infinite loop until an interrupt occurs. In this test case, a previously captured file with .pcap extension will be parsed by the program in order to document the speed of reading and parsing. Test system is running Ubuntu 12.04, has two CPU cores at 1.7 GHz, and the memory is 1 GB. Input .pcap file contains more than two million packets captured over several

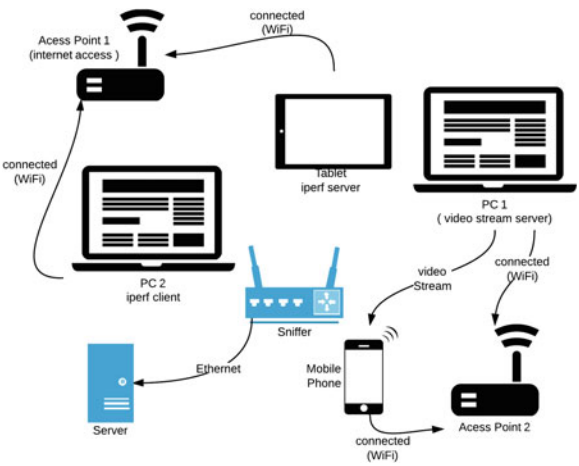


Fig. 2 Test setup with two independent networks

hours. The intention of this test is to see whether the parser is fast enough to handle the high traffic loads or not. Memory consumption is one of the concerns of real time data monitoring. Applications running constantly have to optimize the memory usage. The final version of the pcap parser consumes low memory and does not store data on the memory. By doing so, memory usage is always stable and memory usage for the system only running the parser rarely passed the thirty percent. According to the log, 500,000 packets are parsed in execution time of 1481.1 s that is approximately 24.7 min. This means that parser application can provide real time output for the traffic load of one million packets per hour without any delay. While analyzing the capture files, we have observed that among the 22 million packets, 80 % of the packets had size varied between 20 and 320 bytes.

4 Conclusion and Future Work

Our solution offers a low cost, off the shelf, and customizable alternative to both commercial and open source packet capture systems. User interface provides statistics such as percentages of frame types and subtypes, and the number of frames transmitted by each access point. Since we do not store entire packet contents, our database only contains the practical information about the frames. Our work is expected to be beneficial for academic projects and startups.

For future work, sniffer packet sending performance can be improved. Also, components such as pcap parser and Node.js server can be modified to run on RPi. Another future work plan is to trim the packets in RPi and send the relevant information to the main server. By doing so, number of packets that can be monitored in real time will significantly increase.

References

1. N.T. Anh, R. Shorey, Network sniffing tools for WLANs: Merits and limitations, in *IEEE International Conference on Personal Wireless Communications, 2005. ICPWC 2005*, 23–25 January 2005. doi:10.1109/ICPWC.2005.1431372
2. M.J. Jipping, A. Kalafut, N. Kooistra, K. Ludewig, Investigating wired and wireless networks using a java-based programmable sniffer. SIGCSE Bull. **36**(3), 12–16 (2004), http://doi.acm.org/10.1145/1026487.1008003
3. K.C.Y. Shum, Q.J. Cheng, J.K.Y. Ng, A signal strength based location estimation algorithm within a wireless network, in *IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 22–25 March 2011. doi:10.1109/AINA.2011.80.
4. K.C.Y. Shum, J.K.-Y. Ng, Q.J. Cheng, The design and implementation of a wireless location estimation system in a wireless local area network, in *26th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 26–29 March 2012. doi:10.1109/WAINA.2012.169
5. K.C. Shum, J.K. Ng, Detecting, locating, and tracking hacker activities within a WLAN network, in *16th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 23–25 August 2010. doi:10.1109/RTCSA.2010.46

6. T. Henderson, D. Kotz, I. Abyzov, The changing usage of a mature campus-wide wireless network, in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)* (ACM, New York, USA, 2004), pp. 187–201, http://doi.acm.org/10.1145/1023720.1023739
7. D. Boughaci, K. Ider, S. Yahiaoui, Design and implementation of a misused intrusion detection system using autonomous and mobile agents, in *Proceedings of the 2007 Euro American conference on Telematics and information systems (EATIS '07)* (ACM, New York, USA, 2007), Article 12, p. 8, http://doi.acm.org/10.1145/1352694.1352707
8. R. Henders, B. Opdyke, Detecting intruders on a campus network: might the threat be coming from within? in *Proceedings of the 33rd Annual ACM SIGUCCS Fall Conference(SIGUCCS '05)* (ACM, New York, USA, 2005), pp. 113–117, http://doi.acm.org/10.1145/1099435.1099461
9. http://www.tcpdump.org/ retrieved 15/12/2013
10. http://www.wireshark.org/ retrieved 15/12/2013
11. http://www.wildpackets.com/products/ retrieved 15/12/2013
12. http://wiki.openwrt.org/toh/start retrieved 04/06/2014
13. A.M. Polli, M. Korn, C.N. Klokmose, Local area artworks: collaborative art interpretation on-site, in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp '13 Adjunct)* (ACM, New York, USA, 2013), pp. 79–82. http://doi.acm.org/10.1145/2494091.2494114
14. http://www.raspberrypi.org/ retrieved 22/04/2014
15. http://www.tcpdump.org/pcap.html retrieved 22/04/2014
16. http://wireless.kernel.org/en/users/Documentation/modes retrieved 22/04/2014
17. Y. Turk, *Wireless Sniffer System*. Senior Project Report (Yeditepe University, Istanbul, 2013)
18. http://nodejs.org/ retrieved 04/01/2014
19. http://en.wikipedia.org/wiki/Push_technology retrieved 22/04/2014

Visual Analytics for Enhancing Supervised Attack Attribution in Mobile Networks

Stavros Papadopoulos, Vasilios Mavroudis, Anastasios Drosou and Dimitrios Tzovaras

Abstract Researchers have recently uncovered numerous anomalies that affect 3G/4G networks, caused either by hardware failures, or by Denial of Service (DoS) attacks against core network components. Detection and attribution of these anomalies are of major importance for the mobile operators. In this respect, this paper presents a lightweight application, which aims at analyzing signaling activity in the mobile network. The proposed approach combines the advantages of anomaly detection and visualization, in order to efficiently enable the analyst to detect and to attribute anomalies. Specifically, an outlier-based anomaly detection technique is applied onto hourly statistics of multiple traffic variables, collected from one Home Location Register (HLR). The calculated anomaly scores are afterward visualized utilizing stacked graphs, in order to allow the analyst to have an overview of the signaling activity and detect time windows of significant change in their behavior. Afterward, the analyst can perform root cause analysis of suspicious time periods, utilizing graph representations, which illustrate the high-level topology of the mobile network and the cumulative signaling activity of each network component. Experimental demonstration on synthetically generated anomalies illustrates the efficiency of the proposed approach.

This work has been partially supported by the European Commission through project FP7-ICT-317888-NEMESYS funded by the 7th framework program. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

S. Papadopoulos (✉)
Imperial College London, London, UK
e-mail: s.papadopoulos11@imperial.ac.uk; spap@iti.gr

S. Papadopoulos · V. Mavroudis · A. Drosou · D. Tzovaras
CERTH-ITI, Thessaloniki, Greece
e-mail: mavroudis@iti.gr

A. Drosou
e-mail: drosou@iti.gr

D. Tzovaras
e-mail: tzovaras@iti.gr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_21

1 Introduction

Mobile networks and devices are becoming the targets of cyber criminals aiming to exploit the infrastructure and the provided services for their purposes. As a countermeasure, mobile network operators employ authentication-based techniques to prevent illegitimate users from attaching to the network. Malicious individuals, however, can still infiltrate the network by utilizing compromised mobile devices of legitimate subscribers, and launching attacks against the infrastructure or the subscribers. The main focus of this paper is the detection and attribution of signaling-oriented Denial of Service (SDoS) attacks [1, 2], which target mobile network components in the core network. The effect of a SDoS attack can be amplified when a botnet (i.e., network of compromised devices) is utilized to launch attacks from multiple nodes, which target to overload a specific component of the network.

1.1 Related Work

Anomaly detection techniques for the detection of signaling attacks in 3G/4G networks have been proposed in the literature. Specifically, Lee et al. [2, 3] proposed a cumulative sum (CUSUM)-based method for the detection of signaling attacks that the traditional detection systems cannot detect. The authors designed their method so that is it hard for the attackers to evade detection. They also evaluated their approach against a novel SDoS attack that affects the RNC and the Node-B in 3G and potentially WiMax networks. Alconzo et al. [4] propose statistical techniques applied on time series of unidirectional feature distributions. Coluccia et al. [5] present two distribution-based anomaly detection methods and propose enhancements on the method introduced in [4].

Apart from the analytical methods for anomaly detection, visualization-based methods have also been proposed [6]. For instance, visualizations based on graph representations of the network topology, have been successfully used in network security. Lad et al. [7] proposes a graph representation of the Border Gateway Protocol (BGP) network topology, which illustrates the routing behavior over a specific time period. The volume of the BGP routing changes computed on the graph has been proposed as a descriptive feature that allows for the detection of anomalous time periods. Shi et al. [8] proposed the "SAVE" system, which utilizes graph representations to illustrate the packet delivery paths in sensor networks, based on the so-called GrowthRingMaps [9].

Last but not least, anomaly detection and visualization have been also combined recently, implementing anomaly quantification, such as the power consumption-based anomaly detection in buildings [10] from time series.

1.2 Motivation

To the best of the authors' knowledge, no previous work has addressed security threads in the control plane of mobile networks by combining both information visualization and anomaly detection techniques. Thus, the main motivation of the proposed system is to bridge this gap and provide a system for the visual analysis and detection of signaling-related anomalies.

This paper proposes a novel system for providing an overview of the mobile network signaling activity in suspicious time instances, and for preforming root cause analysis. The main advantage of the proposed system is that it is very lightweight in computational resources. The reason for this is that it operates directly on the statistical data collected from the network with out the need for feature extraction and preprocessing. Additionally, the graph layout is static since it represents the mobile network topology, a fact which eliminates the need for heavy layout computations in the case of structural changes.

The rest of the paper is organized as follows: Section 2 presents the details of the proposed anomaly detection approach. The evaluation takes place at Sect. 3, while the paper concludes at Sect. 4.

2 System Overview

This section presents an overview of the proposed system. It is comprised of two parts, the anomaly detection module and the visualization module. Initially, signaling traffic statistics are collected from the monitoring points in the control plane of the 3G/4G mobile network (Sect. 3.1). Afterward, these signaling data are fed into the anomaly detection module, which utilizes an outlier-based detection method in order to compute anomaly scores for each time period under investigation. The scores are computed by calculating the distance between the examined network traffic instance and the normal traffic instances observed in the past. Thereafter, the visualization module utilizes the anomaly scores, so as to enable the analyst to have an overview of the signaling activity over time, and detect anomalous time periods. Additionally, a graph-based representation of the mobile network, facilitates the task of visualizing the actual signaling behavior of each network component for the selected time period, thus enabling the root cause analysis of the anomalies under investigation.

2.1 Anomaly Detection Module: Identification of Outliers in the Control Plane

2.1.1 Problem Definition

Anomaly detection refers to the identification of network traffic instances that do not conform with normal network behavior [11]. For the definition of the anomaly

detection problem, two matrices are used. The first matrix is matrix D, which serves as ground truth and contains only normal traffic instances and the second matrix is E, which is the input for the anomaly detection method: $D = \{d_{i,j}|$ where $i \in [1, K], j \in [1, Y]\}$, and $E = \{e_{k,l}|$ where $k \in [1, K], l \in [1, Z]\}$. K is the number of traffic variables and Y is the number of observations of normal network traffic and Z is the number of observations that need to be evaluated with regards to their normality. Furthermore, each element of the matrix D is denoted as d_{ij}, the observation sequence of a traffic variable as $D_{\text{row}}(i) = \{d_{ij}, \forall j \in [1, Y]\}$, where $i \in [1, K]$ and a traffic instance as $D_{\text{col}}(j) = \{d_{ij}, \forall i \in [1, \ldots, K]\}$, where $j \in [1, Y]$. In both matrices D and E, each row corresponds to a *traffic variable* and each column to an observed *traffic instance*. Based on these definitions, the anomaly detection problems refers to the detection of the traffic instance $E_{\text{col}}(j)$, which deviate from the normal behavior. It should be underlined that the time intervals between consecutive instances remain the same.

2.1.2 Local Outlier Factor Method

Based on the traffic model outlined in the previous section, each traffic instance is modeled as a point in the K-dimensional space. Subsequently, the local outlier factor (LOF) [12] method is applied, so as to detect any anomalous traffic instances found in E. LOF operates by comparing the spatial density around a given point with the density around its k nearest points and then provide a score which indicates if the examined point resides in a low-density area or not. More formally, $\forall e_{(i,n)} \in E$, the outlier score is computed using equation (1), as defined in [12]. The $lrd(e_{\text{col}}(n))$ function defines the local reachability density of $e_{\text{col}}(n)$ [12].

$$LOF(e_{\text{col}}(n)) = \frac{\sum_{d_{\text{col}}(m) \in R_{(e_{\text{col}}(n))}} lrd(d_{\text{col}}(m))/lrd(e_{\text{col}}(n))}{| R(e_{\text{col}}(n)) |} \tag{1}$$

where $R_{(e_{\text{col}}(n))}$ contains the k-nearest neighbors of $e_{(i,n)}$ from D. More details on the LOF method can be found in [12]. In cases of normal traffic instances, the LOF score is $\sim$1, whereas abnormal instances would exhibit significant deviations from this base [12].

It should be noted that the part of the algorithm which detects the k nearest neighbors is the most computationally demanding. More specifically, for a dataset with stable dimensionality this part has complexity of $O(N^2)$. Hence, in our case the runtime of the LOF algorithm was found to be sufficiently fast for computing outliers in real-time with computation time $<$5 s on each iteration (i.e., for each hour). This can be attributed to the fact that the size and the dimensionality of the ground-truth dataset D do not change.

2.2 Visualization Module: Root Cause Analysis of Signaling Anomalies

2.2.1 An Efficient Approach for Multiple Visualizations Over Extended Periods of Time

In order to draw the attention of the analyst in interesting parts of the data, temporal visualizations of anomaly scores are utilized. This way, the analyst can has an overview of the signaling activity over time, and can detect anomalous time periods and perform a more detailed analysis. The proposed temporal visualizations provide an information rich overview of the data and enable the efficient understanding and exploration of the datasets, following the information seeking mantra suggested by Shneiderman: "*Analyze first, show the important, zoom, filter and analyze further, details on demand*" [13].

The temporal visualization utilized by the proposed approach are depicted in Fig. 1a, b. The first overview is provided by the stacked graph representation. This representation provides an overview of the anomaly scores for one network component, computed for each time period, while it also provides information regarding the normalized volume of each traffic variable for the same period, utilizing the z-score normalization. Each traffic variable is represented using a different color. The glyph representation also provides an overview of the anomaly scores of each time period, for one network component. Color is utilized to represent the actual value of the score at each time period, while the circular layout of the scores represents the time parameter, resembling a clock metaphor. Both these methods are used in combination, in order to provide an information rich visualizations and allow the analyst detect anomalous time periods to focus on, and perform root cause analysis utilizing the Mobile Network Graph presented in the next section.

2.2.2 Visualization of the Network Topology and Signaling Activity

The section presents the Mobile Network Graph visualization approach that is utilized for root cause analysis. The proposed approach utilizes a graph representations, to illustrate the actual topology of the mobile network. Each node represents a network

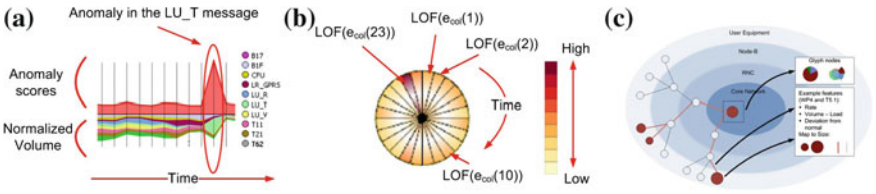


Fig. 1 **a** Stacked graphs visualization of the anomaly scores and signaling volume. **b** Glyph representation of the anomaly scores for one network component. **c** A scheme of the proposed layered layout of the Mobile Network Graph

component, while edges illustrate connections between them. The positioning of the nodes of the graph is calculated in consecutive layers. This layout enables the easier perception of the topology of anomalous events, since one additional visual element, i.e., the position, is utilized to encode additional topological information. Specifically, four layers, comprised of multiple network components are defined, as shown in Fig. 1c:

1. UE (User Equipment): Contains actual mobile devices.
2. NodeB: Contains NodeBs, which are the network component that provides the mobile devices with wireless connectivity. Each NodeB serves multiple mobile devices, i.e., all the devices in range.
3. RNC: Contains RNCs (Radio Network Controller)which are responsible for controlling the NodeBs that are connected to them.
4. Core Network: Core network is the central part of a telecommunication network that is responsible for providing all the network services to the customers, e.g., SMS/Call routing. This layer is comprised of all the nodes that belong to the Core Network, e.g., the most important are SGSN (Serving GPRS Support Node), GGSN (Gateway GPRS Support Node), HLR (Home Location Register), MSC (Mobile Switching Center), and VLR (Visitor Location Register).

It should be noted that the proposed layered graph layout is computed only once, and is thereafter static. The main computational bottleneck of the proposed system is the anomaly detection through the LOF calculation ($O(N^2)$). But as mentioned in Sect. 2.1.2, the LOF implementation is very fast, capable of operatizing in real-time. These facts render the system very lightweight in computational recourses and enable the real-time analysis of signaling anomalies. Screenshots of the developed system are shown in Figs. 2 and 3.

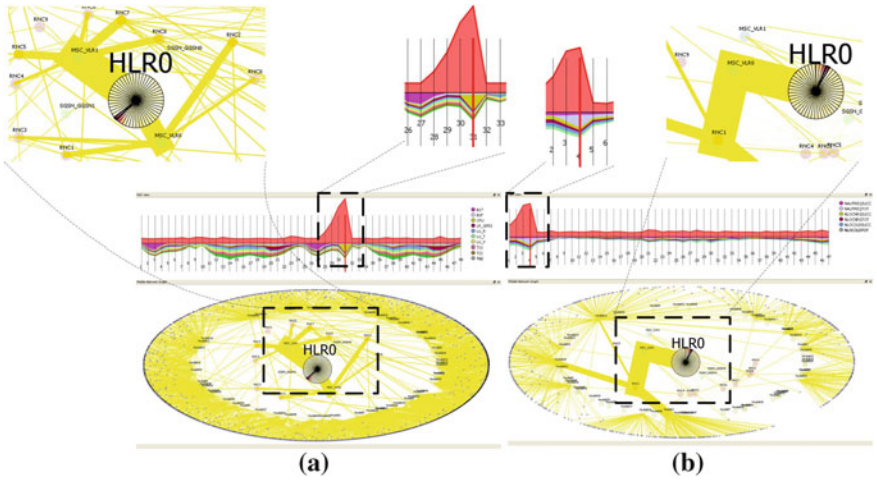


Fig. 2 The network visualization during two abnormal network events, **a** DDos attack **b** an authentication attack

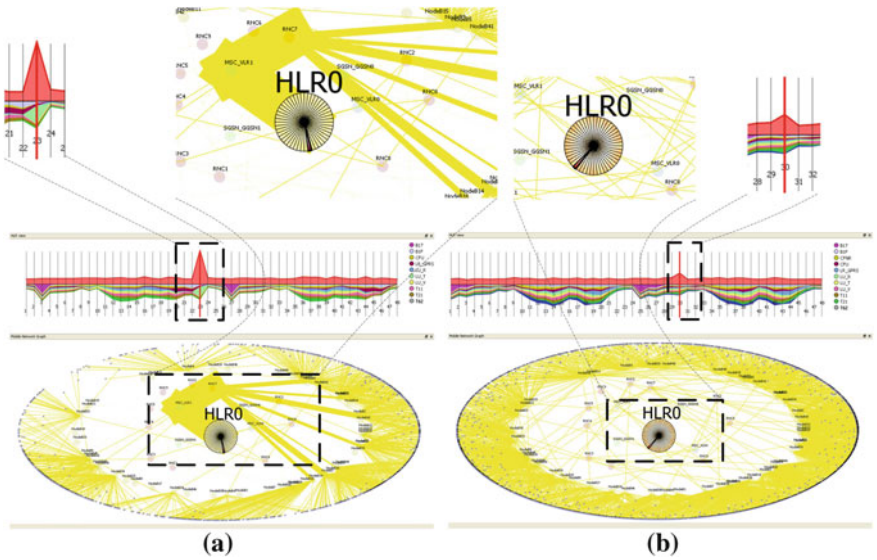


Fig. 3 The network visualization during two abnormal network events, **a** an RNC restart **b** an SMS spam campaign

3 Demonstration on Suspicious Incidents

3.1 Datasets

In order to demonstrate the effectiveness of the proposed system four synthetic datasets were generated using two raw datasets containing traces collected from the 3G/4G mobile networks of two major European telecommunications providers. These raw datasets contain statistical data collected from one HLR of the network. Additionally, they reportedly include no abnormal network incidents, so it is assumed that all the traffic instances included in each dataset exhibit normal network activity. The first raw dataset contains data for 10 signaling messages (e.g., location update etc.) with granularity of one hour. The second raw dataset contains data for three types of signaling requests. However, none of these datasets provide information about the activity of individual subscribers and their position in the network topology (e.g., NodeB that the subscriber is attached to) and thus do not provide enough information for performing root cause analysis of any detected anomalies. To address this, a data generator was designed and implemented. The generation process is defined as follows: (1) Initially the raw data containing the HLR signaling traffic statistics are fed into the generator, (2) the raw data are analyzed and the corresponding Gaussian distribution for each time of the day is computed, (3) then based on the overall traffic for the HLR the traffic distribution for a normal user is computed (Normal User traffic profile), and (4) using the aforementioned profiles a synthetic dataset

which contains statistical data for the HLR and for each subscriber is generated. As a final step, abnormal instances were manually inserted based on the related literature [14–18].

3.2 Demonstration Scenarios

Based on network incidents that have been studied in the literature [14–18], different scenarios were designed and implemented so as to evaluate the proposed approach. These scenarios cover incidents that vary from network component malfunctioning (i.e., Sect. 3.2.3) to malicious attacks (i.e., Sect. 3.2.2). The performance of the proposed approach is demonstrated in the sections that follow.

3.2.1 Call Forwarding DDoS

This scenario simulates a DDoS attack that has been proposed by Traynor et al. in [15] and can effectively overload the HLR/HSS component, so as to degrade the QoS for the network subscribers. More specifically, a large number of mobile devices are compromised by malware and launch a DDoS attack by exploiting call forwarding signaling requests. Although, these requests on a per packet basis are legitimate, such a coordinated attack can significantly increase the load in the HLR/HSS and render it unresponsive. In this scenario, the attack has a high growth rate and reaches its peak (i.e., maximum throughput from infected devices) very fast. As seen in Fig. 2a, our proposed approach is able to detect and highlight the incident from an early stage and visually inform the human operator before the network stability is affected. As shown in the stacked graph representations, at the time of the anomaly, the CFU (Call Forwarding Unconditional) signaling message has increased volume. Upon the detection, the operator is able to clearly identify the distributed nature of the attack, since the excessive signaling does not originates from a specific region of the network, but instead it is uniformly distributed. The operator also is able to identify the exploited signaling messages and the misbehaving subscribers, and design countermeasures (e.g., rejection rules for the control plane firewall).

3.2.2 Authentication Attack

In this scenario, an authentication attack which uses sim-less devices, as described in [14], is implemented. As in Sect. 3.2.1, the attacker aims to affect the network stability and degrade the performance of the network. However, in this case the attacker is regionally constrained and thus he launches the attack from a limited number of NodeBs which all belong in the same geographical area. During the attack, the proposed approach initially detects an abnormal increase in the incoming authentication attempts and displays a high anomaly score. The human operator,

then uses the stacked graph to compare the volume of authentication attempts with the volume of successful authentication transactions and verifies that the observed behavior is potentially malicious, since the success/attempts ratio is very low. Then he/she utilizes the network graph to identify the root cause of the incident. As seen in Fig. 2b, RNC1 exhibits increased traffic load, due to the high volume of signaling messages received from specific NodeBs (i.e., NodeB1, NodeB10, NodeB11). This analysis enables the operator, which know is aware of the nature of the attack, to design effective mitigation countermeasures. For instance, one mitigation approach would be to lower the processing priority of authentication requests originating from the affected NodeBs in order to avoid congestion in the HLR/HSS.

3.2.3 RNC Restart

In this scenario, the proposed approach is evaluated during the restart of a malfunctioning RNC [18]. According to [18] the network becomes unstable due to the flood of location update requests toward the HLR/HSS, as the mobile devices switch from 2G to 3G or 4G, when the malfunctioning component becomes available again. More specifically, as seen in Fig. 3a, the RNC7 has been restarted and this results in a flood of signaling requests from all subscribers which reside in the NodeBs it controls. Our proposed approach displays an increased anomaly score and informs the human operator of that the volume of location update requests from home subscribers (i.e., LU_T) is higher than normal. Additionally, the network graph provides enough information for the operator to pin point the root cause of the incident, i.e., RNC7.

3.2.4 Spam SMS Campaign

This scenario was designed based on the findings of [16, 17, 19] regarding the modus operandi of the spammers and the impact of their activities on the network. In particular, a spam outburst was implemented as an increase in the number of mobile terminated SMS messages toward the users of the monitored HLR/HSS was inserted in the dataset. In this scenario, the unsolicited spam messages affect 1 % of the subscribers [16]. This abnormal increase in the number of incoming messages, is detectable from the HLR/HSS component, as it controls all the communications of the subscribers it serves. The proposed approach, depicted in Fig. 3b, initially detects the anomalous incident by examining the incoming traffic of the HLR/HSS and displays a small increase in the anomaly score and the volume of the signaling message that relates with mobile terminated SMS messages (i.e., Send Routing Information T21). In the specific scenario, the spam messages originate from an attacker that resides outside of the MNOs network and thus the network graph does not indicate any significant increase in the network volume of the NodeB, RNC, and MSC/SGSN components.

4 Conclusions

A complete system for anomaly detection and root cause analysis in the mobile network has been presented. The proposed system enables supervised attack attribution and root cause analysis of anomalous phenomena in the mobile network. To achieve this efficiently, it combines methods from the fields of anomaly detection and information visualization, in order to enhance the analytical potential and allow the user to understand and explore the data. The efficiency of the proposed approach in detecting and attributing unknown anomalies has been demonstrated based on four network incident scenarios that affect the currently deployed mobile networks and have been analyzed in depth in the literature. The current implementation of the system enables the human operator to analyze anomalies in the control plane of the mobile network. Future work includes the enhancement of the current system in order to visualize both the user plane and the control plane, and further improve the root cause analysis of events, such as signaling storms. User studies will also be performed in order to evaluate the proposed approach on real users, including mobile network analysts.

References

1. G. Kambourakis, C. Kolias, S. Gritzalis, J.H. Park, DoS attacks exploiting signaling in UMTS and IMS. Comput. Commun. **34**(3), 226–235 (2011)
2. P.P.C. Lee, T. Bu, T. Woo, On the detection of signaling DoS attacks on 3G wireless networks, in: *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1289–1297, 2007.
3. P.P.C. Lee, T. Bu, T. Woo, On the detection of signaling DoS attacks on 3G/WiMax wireless networks. Comput. Netw. **53**(15), 2601–2616 (2009)
4. A. D'Alconzo, A. Coluccia, F. Ricciato, P. Romirer-Maierhofer, A distribution-based approach to anomaly detection and application to 3G mobile traffic, in: Global Telecommunications Conference, GLOBECOM 2009. IEEE, pp. 1–8, 2009.
5. A. Coluccia, A. DAlconzo, F. Ricciato, Distribution-based anomaly detection in network traffic, in: Data Traffic Monitoring and Analysis, Springer, pp. 202–216, 2013.
6. H. Shiravi, A. Shiravi, A.A. Ghorbani, A survey of visualization systems for network security. IEEE Trans. Vis. Comput. Graph. **1**(1), 1–19 (2011)
7. M. Lad, D. Massey, L. Zhang, Visualizing internet routing changes. IEEE Trans. Vis. Comput. Graph. **12**(6), 1450–1460 (2006)
8. L. Shi, Q. Liao, Y. He, R. Li, A. Striegel, Z. Su, SAVE: Sensor anomaly visualization engine, in: IEEE Conference on Visual Analytics Science and Technology (VAST), IEEE, pp. 201–210, 2011.
9. G. Andrienko, N. Andrienko, P. Bak, D. Keim, S. Kisilevich, S. Wrobel, A conceptual framework and taxonomy of techniques for analyzing movement. J. Vis. Lang. Comput. **22**(3), 213–232 (2011)
10. H. Janetzko, F. Stoffel, S. Mittelstädt, D.A. Keim, Anomaly detection for visual analytics of power consumption data. Comput. Graph. **38**, 27–37 (2014)
11. V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey. ACM Comput. Surv. (CSUR) **41**(3), 15 (2009)
12. M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: ACM Sigmod Record, vol. 29, pp. 93–104, ACM, 2000.

13. B. Shneiderman, The eyes have it: a task by data type taxonomy for information visualizations, in: Proceedings of the 1996 IEEE Symposium on Visual Languages, VL '96, 1996.
14. N. Gobbo, A. Merlo, M. Migliardi, A denial of service attack to GSM networks via attach procedure, in: Security Engineering and Intelligence Informatics, Springer, pp. 361–376, 2013.
15. P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, T. La Porta, On cellular botnets: measuring the impact of malicious devices on a cellular network core, in: Proceedings of the 16th ACM conference on Computer and communications security, pp. 223–234, ACM, 2009.
16. N. Jiang, Y. Jin, A. Skudlark, Z.-L. Zhang, Understanding sms spam in a large cellular network: characteristics, strategies and defenses, in: Research in Attacks, Intrusions, and Defenses, Springer, pp. 328–347, 2013.
17. T.A. Almeida, J.M.G. Hidalgo, A. Yamakami, Contributions to the study of sms spam filtering: new collection and results, in:textitProceedings of the 11th ACM Symposium on Document Engineering, pp. 259–262, ACM, 2011.
18. 3GPP, Study on Core Network Overload (CNO) Solutions, TS 23.843, 3rd Generation Partnership Project (3GPP), 12 2013.
19. S.J. Delany, M. Buckley, D. Greene, Sms spam filtering: methods and data. Expert Syst. Appl. **39**(10), 9899–9908 (2012)

Part V
Queuing Systems

Quantum Queuing Networks Throughput Optimisation

Dariusz Kurzyk and Piotr Gawron

Abstract We study models of quantum queues based on discrete time quantum walks with barriers. Our considerations refer to multi-servers queuing models. Input and output of jobs in the queue are realised by systems consisting of quantum coins and walkers. We show that presented models behave differently from the classical ones. We also present numerical methods for optimisation of jobs processed by the system. We simultaneously maximise jobs throughput in the system and minimise the number of jobs lost.

1 Introduction

Queuing models are often modelled using Markov processes. When numerical methods are considered for analysis of queuing systems, the state space of the modelled system is discrete and transitions are given by stochastic matrices [1]. Thus, these kinds of models can be understood as random walks over a graph having complicated structure.

Discrete time Markov chains (DTMC) and random walks can be generalised by discrete time quantum Markov chains (DTQMC) and quantum walks. It means that DTQMC are reducible to DTMC and quantum queuing model are more general than the classical models. In quantum information and computation theory, there exists a notion of quantum walk. Usually by quantum walk one understands a pair consisting of an unitary operator and initial quantum state. Time evolution of such system is given by sequential application of unitary operator on the quantum state [4]. Quantum walks have found many applications as quantum algorithms or models of physical

D. Kurzyk (✉)
Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
Bałtycka 5, 44-100 Gliwice, Poland
e-mail: dkurzyk@iitis.pl

P. Gawron · D. Kurzyk
Institute of Mathematics, Silesian University of Technology,
Kaszubska 23, 44-100 Gliwice, Poland

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_22

quantum systems. In this work, we consider quantum walks with absorbing barriers [2] that mimic behaviour of a simple queuing model.

In a recent work [3], quantum queues were modelled using quantum channels and mixed states. In this work, we consider unitary evolution interweaved with quantum measurement with post-selection.

2 Mathematical Preliminaries

Throughout this paper, we use complex Euclidean space. According to Dirac notation, symbol $|\psi\rangle$ called as *ket*, denotes a complex column vector. The conjugate transpose of a *ket* is denotes by *bra* $\langle\psi|$, thus $|\psi\rangle^\dagger = \langle\psi|$. In n-dimensional complex Euclidean space $\mathbb{C}^n$, the *standard orthogonal basis* is expressed by the vectors:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, ..., |n-1\rangle = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}. \tag{1}$$

Every vector $|\psi\rangle \in \mathbb{C}^n$ can be expressed as the linear combination

$$|\psi\rangle = \sum_{i=0}^{n-1} \alpha_i |i\rangle, \tag{2}$$

where α_i are complex number, which satisfy $\sum_{i=1}^{n-1} |\alpha_i|^2 = 1$.

Consider two vectors $|\psi\rangle = \sum_{i=0}^{n-1} \alpha_i |i\rangle$ and $|\phi\rangle = \sum_{i=0}^{n-1} \beta_i |i\rangle$. The *inner product* (*scalar product*) between $|\psi\rangle$,$|\phi\rangle$ is denoted by

$$\langle\psi|\phi\rangle = \sum_{i=0}^{n-1} \alpha_i^* \beta_i. \tag{3}$$

The *outer product* between $|\psi\rangle$,$|\phi\rangle$ is expressed as

$$|\psi\rangle\langle\phi| = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \alpha_i \beta_i^* |i\rangle\langle j|. \tag{4}$$

The *tensor product* between $|\psi\rangle$,$|\phi\rangle$ is defined as the n^2 dimensional vector

$$|\psi\rangle \otimes |\phi\rangle = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \alpha_i \beta_i |i\rangle \otimes |i\rangle. \tag{5}$$

3 Discrete Time Quantum walks

Consider complex Euclidean space $\mathbb{C}^c \otimes \mathbb{C}^p$, where $|\phi\rangle \in \mathbb{C}^c$ is a state of coin and $|\psi\rangle \in \mathbb{C}^p$ describe the position of the walker. Thus $\mathbb{C}^c$, $\mathbb{C}^p$ are coin space and position space, respectively. Each step of the walk is performed by the unitary operation $U = (C \otimes \mathrm{I}) \cdot S$, where C is unitary operator and

$$S = \sum_{i=0}^{c-1} (|i\rangle\langle i| \otimes \sum_{j=0}^{p-1} |n_i(j)\rangle\langle j|), \tag{6}$$

where $n_i(\cdot)$ are permutation functions. After t steps, the state of the walk is $|\psi_t\rangle = U^t|\psi_0\rangle$, where $|\psi_0\rangle$ is an initial state.

4 Quantum Measurement-Driven Evolution

Consider following set of operators $\mathbf{A} = \{A_1, A_2\}$, where

$$A_1 = \sum_{i=0}^{n-k-1} |i+k\rangle\langle i|, \quad A_2 = \mathrm{I}_n - A_1^\dagger A_1, \tag{7}$$

where I_n is n-dimensional identity matrix. The constant k fulfils $1 \leq k < n-1$. Products of $A_1^\dagger A_1$ and $A_2^\dagger A_2$ can be expressed as $A_1^\dagger A_1 = \sum_{i=0}^{n-k-1} |i\rangle\langle i|$ and $A_2^\dagger A_2 = \sum_{i=n-k}^{n-1} |i\rangle\langle i|$, respectively. Thus, it is easy to check that

$$A_1^\dagger A_1 + A_2^\dagger A_2 = \mathrm{I}_n. \tag{8}$$

Thus, $\mathbf{A}$ satisfies completeness relation and it forms a quantum measurement. Let $\rho = |\psi\rangle\langle\psi|$ be an initial state, then measurement performed on the initial state transforms ρ into ρ_i and gives ith outcome with probability p_i [5] according to

$$\rho \to \rho_i = \frac{A_i \rho A_i^\dagger}{\mathrm{tr}(A_i \rho A_i^\dagger)} \quad \text{with} \quad p_i = \mathrm{tr}(A_i \rho A_i^\dagger). \tag{9}$$

If we consider $\rho = |j\rangle\langle j|$, where $|j\rangle \in \mathbb{C}^n$ is a state from standard basis and $j < n-k$, then selective measurement performed by A_1 transforms ρ into $\rho_1 = |j+k\rangle\langle j+k|$. Hence proposed measurement causes incrementation of state ρ without overflow. Similarly, we can perform the measurement based on operators $\mathbf{B} = \{B_1, B_2\}$, where

$$B_1 = \sum_{i=k}^{n-1} |i-k\rangle\langle i|, \quad B_2 = \mathrm{I}_n - B_1^\dagger B_1. \tag{10}$$

The set **B** satisfies completeness relation and selective measurement B_1 performed on the $\rho = |j\rangle\langle j|$ where $k \leq j < n$ transforms ρ to $\rho_1 = |j-k\rangle\langle j-k|$. Thus, the measurement with respect to operator B_1 causes decrementation of ρ without underflow.

Next, we construct controlled selective measurement in following way. Let ρ be a state given by $\rho = |\psi\rangle\langle\psi| \otimes |\phi\rangle\langle\phi|$, where $|\psi\rangle$, $|\phi\rangle$ are n_1 and n_2 dimensional, respectively. If $|\psi\rangle\langle\psi| = |c\rangle\langle c|$, then we assume that measurement is performed on second register of ρ. Controlled measurement operators are expressed as:

$$A_1^c = |c\rangle\langle c| \otimes A_1 + (\mathrm{I}_{n_1} - |c\rangle\langle c|) \otimes \mathrm{I}_{n_2}, \quad A_2^c = \mathrm{I}_{n_1 n_2} - (A_1^c)^\dagger A_1^c. \tag{11}$$

In the case of controlled selective measurement B_1 on first register of ρ with the condition that $|\phi\rangle\langle\phi| = |c\rangle\langle c|$, there is obtained following operators

$$B_1^c = B_1 \otimes |c\rangle\langle c| + \mathrm{I}_{n_1} \otimes (\mathrm{I}_{n_2} - |c\rangle\langle c|), \quad B_2^c = \mathrm{I}_{n_1 n_2} - (B_1^c)^\dagger B_1^c. \tag{12}$$

In the next following sections, presented measurements will be used to construct models of quantum queues and will be denoted by $\mathcal{M}_+$ and $\mathcal{M}_-$, where

$$\mathcal{M}_+(\rho) = \frac{A_i^c \rho (A_i^c)^\dagger}{\mathrm{tr}(A_i^c \rho (A_i^c)^\dagger)} \quad \text{and} \quad \mathcal{M}_-(\rho) = \frac{B_i^c \rho (B_i^c)^\dagger}{\mathrm{tr}(B_i^c \rho (B_i^c)^\dagger)}. \tag{13}$$

5 Quantum Queue

Quantum queuing models, presented in the following section, are based on the concept of discrete time quantum walks (DTQW). Models of DTQW are usually realised by quantum coin and walker represented by two coupled quantum systems. The evolution of one step of DTQW is performed by two operations. First operation is realised by application of a unitary operation on coin subsystems. Next, conditional selective measurement on second subsystem related with walker is performed. Our models of quantum queues are based on the idea, that quantum coins represent the flow of jobs into and out of the queuing system, similarly the walker's position represents the queue occupancy. We assume that considered queuing systems are finite, thus the models are realised by DTQW with barriers. This kind of DTQW is possible to implement using measurements with post-selection. The lower barrier and upper barrier are interpreted as sinks for jobs lost due to overflow and that jobs processed, respectively.

The typical quantum coins, considered in the literature are

$$\mathcal{DFT} = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^n - 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{1(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{bmatrix} \tag{14}$$

and

$$\mathcal{G}_{a,b} = \begin{bmatrix} a & b & b & \dots & b \\ b & a & b & \dots & b \\ b & b & a & \dots & b \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b & b & b & \dots & a \end{bmatrix}, \tag{15}$$

where a and b are real, $1 - \frac{2}{n} \leq |a| \leq 1$ and $b = \pm(1 - a)$.

Consider quantum system consisting of tree subsystems ψ_1, ϕ, ψ_2, where $|\psi_1\rangle$, $|\phi\rangle$ and $|\phi_2\rangle$ are 2, 64 and 2 dimensional, respectively. First and third registers are coin subsystems related with increasing and decreasing selective measurements, respectively. Second register is associated with walker dependent on two coins. The state of the queue ρ_t can be expressed as $\rho_t = \mathcal{M}_-(\mathcal{C}\mathcal{M}_+(\mathcal{C}\rho_{t-1}))$, which can be presented by schematic representation shown in Fig. 1.

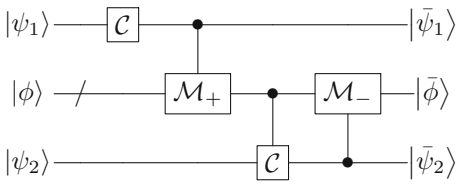


Fig. 1 Schematic representation of DTQW dependent on two coins

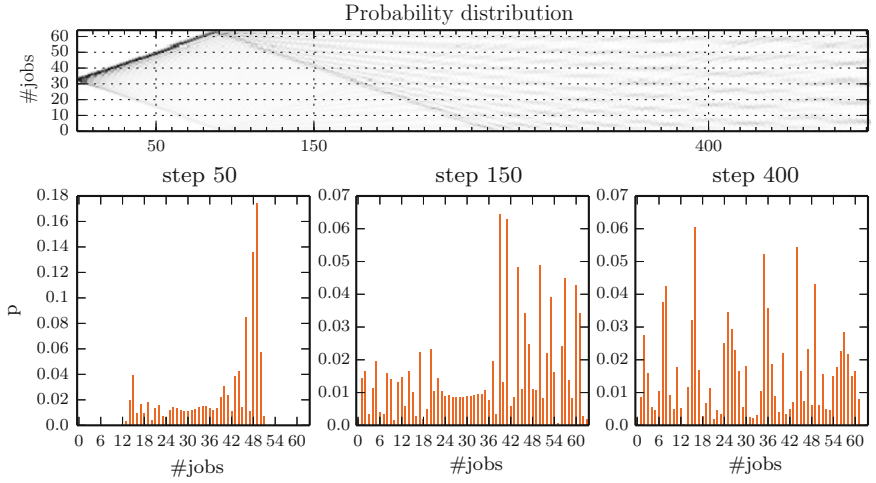


Fig. 2 Probability of the queue occupancy

Example 1 As an example, we can take coins given by DFT and initial state $\rho_0 = |\psi_1\rangle\langle\psi_1| \otimes |\phi\rangle\langle\phi| \otimes |\psi_2\rangle\langle\psi_2| = |0\rangle\langle 0| \otimes |32\rangle\langle 32| \otimes |0\rangle\langle 0|$. Figure 2 shows the probability distributions of the queue occupancy after 500 steps.

6 Quantum Queuing Networks

We consider a simple network consisting of several interconnected workers preceded by associated queues. The workers are connected in sequence. In this network, controlled selective measurements cause incrementation and decrementation of states of subsystems associated with workers that control the flow of the jobs in the system.

Hence, the queues are connected by measurement $\{C_1^c, C_2^c\}$, where

$$\begin{aligned} C_1^c &= B_1 \otimes |c\rangle\langle c| \otimes A_1 + \mathrm{I}_{n_1} \otimes (\mathrm{I}_{n_2} - |c\rangle\langle c|) \otimes \mathrm{I}_{n_3} \\ C_2^c &= \mathrm{I}_{n_1 n_2 n_3 - 1} - (C_1^c)^\dagger C_1^c. \end{aligned} \tag{16}$$

Proposed measurement will be denoted by $\mathcal{M}_\pm(\rho) = \frac{C_i^c \rho (C_i^c)^\dagger}{\mathrm{tr}(C_i^c \rho (C_i^c)^\dagger)}$. The state of the network consisting of two queues can be expressed as

$$\rho_t = \mathcal{M}_-(\mathcal{C}_3 \mathcal{M}_\pm(\mathcal{C}_2 \mathcal{M}_+(\mathcal{C}_1 \rho_{t-1}))), \tag{17}$$

which can be present in a schematic representation shown in Fig. 3.

Example 2 As an example, we consider that $\mathcal{C}$ are DFT coins. The initial state of queuing networks are set in following way. Coin subsystems are in the initial state $|0\rangle\langle 0|$ and worker subsystems are initially set to $|n/2\rangle\langle n/2|$, where n is the dimension of subsystems denoting the queues.

Figure 4 represents the probability distribution of queues occupancies in the case of two queues.

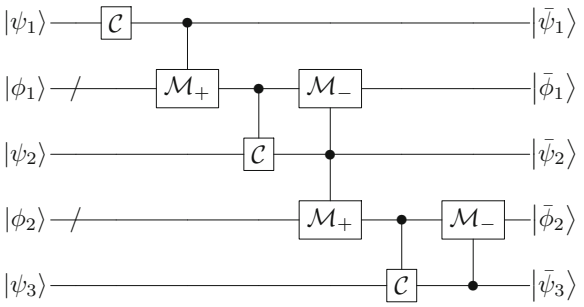


Fig. 3 Schematic representation of quantum queuing network dependent on three coins

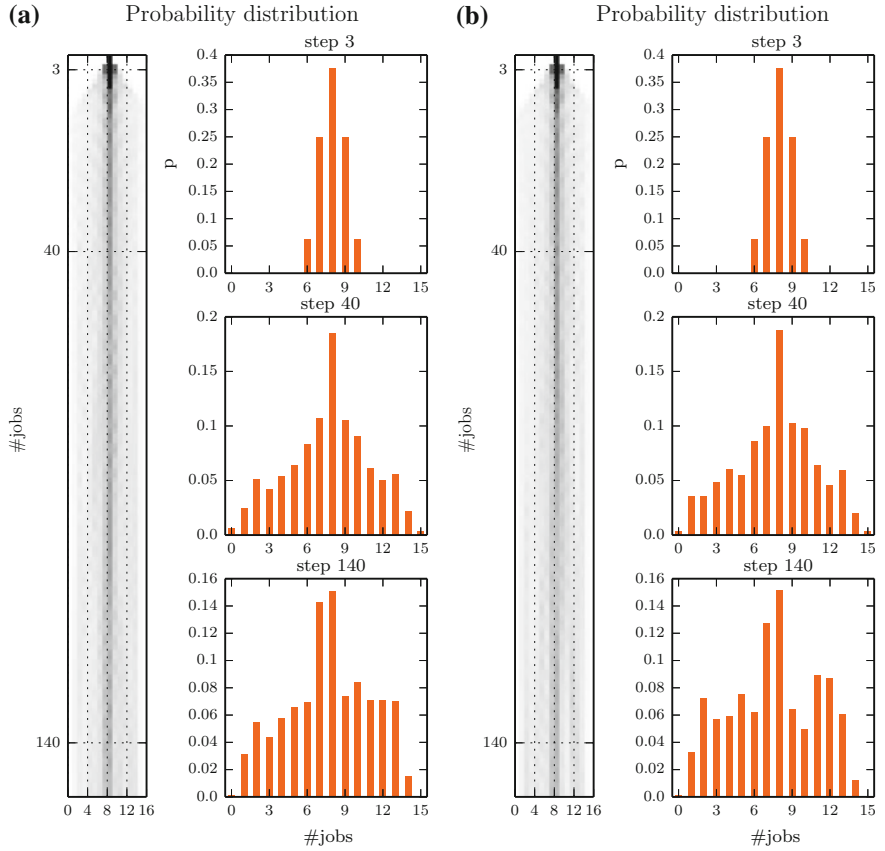


Fig. 4 Probability distribution of the queues occupancies. Panel **a** first queue, panel **b** second queue

7 Throughput Optimisation

The results presented in previous sections show that the usage of DFT as coins causes that the intensities of flow of jobs into and out of the system are similar. Hence, both the probabilities of losing a job and processing of a job are relatively low. We consider models of quantum queuing networks based on DTQW, where in order to maximise the throughput and minimise the losses, the relevant probabilities are optimised. For this purpose, we assume that the coin is realised by $SU(2)$ operation parametrised by three real numbers in the following way:

$$\mathcal{R}_{\gamma,\delta,\theta} = \begin{bmatrix} \mathrm{e}^{-\mathrm{i}(\gamma+\delta)/2}\cos\theta & -\mathrm{e}^{-\mathrm{i}(\gamma-\delta)/2}\cos\theta \\ \mathrm{e}^{\mathrm{i}(\gamma-\delta)/2}\cos\theta & \mathrm{e}^{\mathrm{i}(\gamma+\delta)/2}\cos\theta \end{bmatrix}, \tag{18}$$

where $\gamma, \delta \in [0, 2\pi]$, $\theta \in [-\pi, \pi]$. The goal is to find the parameters γ, δ and θ, where probabilities of jobs being processed are maximised and probabilities of losing jobs are minimised.

Next, we adopt a definition of concurrent hitting time [6].

Definition 1 Quantum walk has (T, p) concurrent (ρ_0, ρ_v) hitting time if the walk with initial state ρ_0 and absorbing boundary at position ρ_v has probability $\geq p$ of stopping at time $t \leq T$, that is $\sum_{t \leq T} p_t \geq p$, where p_t is a probability that the state is absorbed at time t.

In the case of queuing network consisting of two queues, we determine following values

$$p_t = \langle 0|\mathcal{M}'_-(\mathcal{C}_3\mathcal{M}_\pm(\mathcal{C}_2\mathcal{M}_+(\mathcal{C}_1\rho_{t-1})))|0\rangle \tag{19}$$

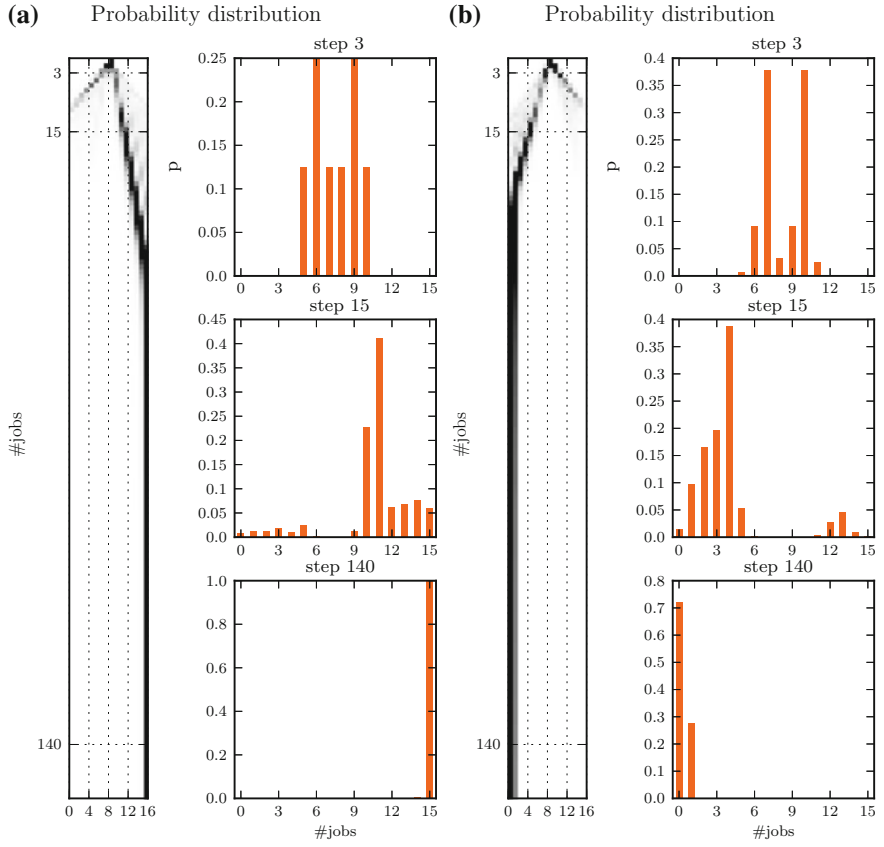


Fig. 5 Probability distributions of queues occupancies for coins with parameters $(\gamma_1, \delta_1, \theta_1, \gamma_2, \delta_2, \theta_2) = (4.2147, 5.2101, 0.0, 2.4042, 5.2027, -1.1052)$

and

$$\hat{p}_t = \langle n_1 - 1|\mathcal{M}'_+(\mathcal{C}_1\mathcal{M}'_-(\mathcal{C}_3\mathcal{M}_\pm(\mathcal{C}_2\rho'_{t-1})))|n_1 - 1\rangle, \tag{20}$$

where $\rho'_{t-1} = \mathcal{M}_+(\mathcal{C}_1\rho_{t-1})$ and $\mathcal{M}'_-$, $\mathcal{M}'_+$ are measurements with respect to operators $\mathrm{I}_2 \otimes B_2^c$, $A_2^c \otimes \mathrm{I}_2$, respectively. Values of p_t are probabilities that jobs are processed and $\hat{p}_t$ are probabilities of losing jobs (it means that jobs are moved from first queue to second queue). Thus, the probability of stopping of the walk at required state is expressed as:

$$\sum_{t \le T} p_t + \hat{p}_t. \tag{21}$$

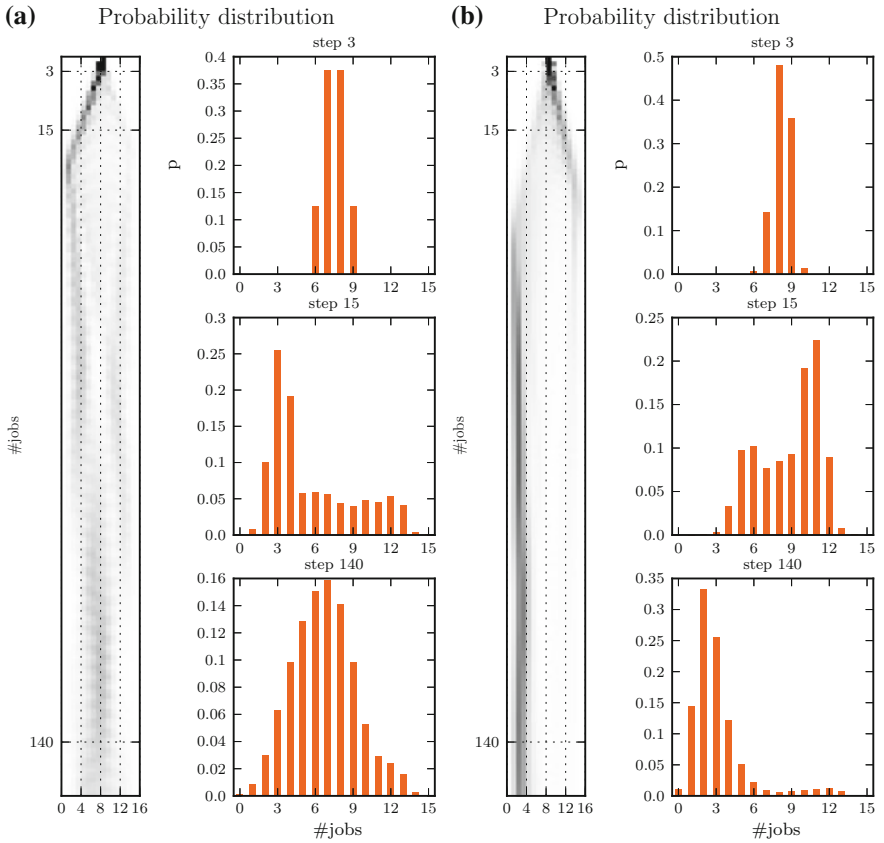


Fig. 6 Probability distributions of queues occupancies for coins $\mathcal{C}_1 = \mathcal{DFT}$, $\mathcal{C}_2 = \mathcal{G}_{0.1021,0.8979}$, and $\mathcal{C}_3 = \mathcal{R}_{4.2147,5.2101,0.0}$

We fix a value of T and we maximise (21). The global optimisation method we use is simulated annealing.

Example 3 Consider a queuing network consisting of two $n_1 = 16, n_2 = 16$ dimensional queues as in Fig. 3. We assume that subsystem related with coins are 2-dimensional. We set initial state as:

$$\rho_0 = \mathcal{H}|0\rangle\langle 0|\mathcal{H}^\dagger \otimes |8\rangle\langle 8| \otimes \mathcal{H}|0\rangle\langle 0|\mathcal{H}^\dagger \otimes |8\rangle\langle 8| \otimes \mathcal{H}|0\rangle\langle 0|\mathcal{H}^\dagger, \tag{22}$$

where $\mathcal{H}$ is Hadamard transform. We also assume that $\mathcal{C}_1$ is DFT coin and $\mathcal{C}_2$, $\mathcal{C}_3$ are $SU(2)$ operations dependent on γ_1, δ_1, θ_1 and γ_2, δ_2, θ_2. After each step, we determine values p_t, $\hat{p}_t$ and next we maximise value of sum expressed in (21) for $T = 150$. As a result of optimisation, we have obtained the following parameters $(\gamma_1, \delta_1, \theta_1, \gamma_2, \delta_2, \theta_2) = (4.2147, 5.2101, 0.0, 2.4042, 5.2027, -1.1052)$, what has been presented in Fig. 5.

Example 4 Next, we set coins $\mathcal{C}_1 = \mathcal{DFT}$, $\mathcal{C}_2 = \mathcal{G}_{a,b}$, $\mathcal{C}_3 = \mathcal{R}_{4.2147,5.2101,0.0}$ and we try find the optimal parameters a, b. As a result of optimisation we get parameters $a = 0.1021$, $b = 0.8979$ (Fig. 6).

8 Results Discussion

We have introduced models for quantum queueing networks modelled as controlled quantum walks with barriers. We have shown that models of quantum queues behave in a different way from their classical counterparts. We have also shown that simulated annealing can be used to maximise the throughput of quantum queues and minimise the jobs loses.

Acknowledgments Acknowledgements Research supported by the Grant N N516 481840 financed by Polish National Science Centre.

References

1. V. Anisimov, *Switching Processes in Queueing Models* (Wiley, New York, 2010)
2. E. Bach, S. Copersmith, M.P. Goldschen, R. Joynt, J. Watrous, One-dimensional quantum walks with absorbing boundaries. J. Comput. Syst. Sci. **69**(4), 562–592 (2004)
3. P. Gawron, D. Kurzyk, Z. Puchała, A model for quantum queue. Int. J. Quantum Inf. **11**(02), 135401 (2013)
4. J. Kempe, Quantum random walks: an introductory overview. Contemp. Phys. **44**(4), 307–327 (2003)
5. M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000)
6. S. Ying, M. Ying, Removing measurements from quantum walks. Phys. Rev. A **87**(1), 012337 (2013)

A Queueing System with Probabilistic Inhomogeneous Vacations for Modeling Power-Saving in Wireless Systems with Retransmissions

Ioannis Dimitriou

Abstract In this paper, we investigate the power management of mobile devices using a variant of an M/G/1 queue with probabilistic inhomogeneous multiple vacations and generalized service process. Under the vacation scheme, at the end of a vacation the server goes on another vacation, with a different probability distribution, if during the previous vacation there have been no arrivals. The modified vacation policy depends on the initial vacation interval and the server selects randomly over M such vacation policies. The theoretical system can be applied for modeling the power saving mode of mobile devices in modern wireless systems. Moreover, the form of the service process properly describes the incremental redundancy retransmission scheme that provides different types of retransmissions in such systems. Steady state analysis is investigated, energy and performance metrics are obtained and used to provide numerical results that are also validated against simulations.

1 Introduction

Power save/sleep mode operation is the key point for the energy efficient usage of battery operated mobile stations (MS). A MS operating in power save mode not only saves the battery energy and enhances lifetime but also introduces unwanted delay in serving data packets arriving during the sleep period. Though energy is a major aspect for handheld devices, delays may also be crucial for time sensitive applications.

Queueing models with vacations [8] have been proved to be useful for the performance analysis of the sleep mode operation in wireless networks. Such models were used ([4–7, 13], not exhausted list) in the analysis of power save methods in modern wireless standards such as 3GPP LTE and IEEE 802.16e. In this work, we consider a general queueing model that properly describes various sleep policies such as the DRX in LTE [1], and the power save classes (PSC) as defined in IEEE 802.16e [2].

I. Dimitriou (✉)
MAESTRO project, INRIA Sophia Antipolis, Sophia Antipolis, France
e-mail: ioannis.dimitriou@inria.fr

Department of Mathematics, Aristotle University of Thessaloniki, Thessaloniki, Greece
e-mail: idimitri@math.auth.gr; idimit@cc.uoi.gr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_23

We assume that the device chooses with a certain probability among a number of available power save policies, and thus, several mixed policies can be applied by varying the model's parameters that may result in better power improvements. In this work, a sleep mode selection policy is employed in selecting among a prularity of sleep schemes. The problem of optimal selection of a sleep policy in IEEE 802.16e among a set of available policies has been addressed using Markov decision process framework in [10].

Clearly, wireless communication is fragile and retransmission schemes are used to cope with this problem. These schemes employ forward error coding (FEC), which is a technique used for controlling errors in data transmission. The joint use of Automatic Repeat reQuest (ARQ), and the FEC algorithm at the transmitter and/or receiver is also employed. Several ways of combining retransmission and channel error coding exist. The incremental redundancy scheme, which is employed in this work, encodes the first transmission with a high rate (and thus low overhead but low protection) while the following transmissions consist of additional redundancy in order to decrease the code rate seen by the receiver. If a maximum number of different retransmissions is reached, Hybrid ARQ (HARQ) declares a failure, and leaves it up to ARQ running in radio link control (RLC) to try again.

The paper is organized as follows. In Sect. 2, we describe the proposed model in detail while in Sect. 3 we provide some useful for the analysis results. The embedded Markov chain analysis is presented in Sect. 4. Using the regenerative approach we obtain interesting performance and energy measures in Sect. 5 that are used to provide numerical results in Sect. 6 that are also valideted against simulations. Moreover, a trade-off between our mixed sleep policy model and other related models which employ standard sleep policies (e.g., DRX in LTE) is investigated.

2 The Model

Packets arrive at the infinite capacity buffer of the base station (BS) according to a Poisson process with rate λ. Then, the BS forwards the packets to the MS. The service process (i.e., the delivery of a packet from the BS towards the MS) consists of at most two cycles. The first service cycle describes the *incremental redundancy* HARQ scheme that take place in the MAC layer, and the second service cycle the ARQ scheme in the RLC layer. The ARQ scheme is requested when a maximum number of retransmissions is reached in the HARQ process.

Assume that the maximum number of retransmissions in the HARQ process is N. The "service operation" is as follows: upon the completion of the ith (re)transmission which needs a time period B_i, $i = 1, \ldots, N-1$ and is arbitrarily distributed with cumulative distribution function (cdf) $B_i(x)$, probability density function (pdf) $b_i(x)$, Laplace-Stieltjes Transform (LST) $\beta_i^*(s)$ and finite moments $\overline{b}_i$, $\overline{b}_i^{(2)}$, the MS succesfully receives the packet with probability q_i whereas with probability $\overline{q}_i = 1 - q_i$ declares an additional retransmission B_{i+1} (see Fig. 1). If N consecutive service times are not satisfactory, the server switches to the second service cycle.

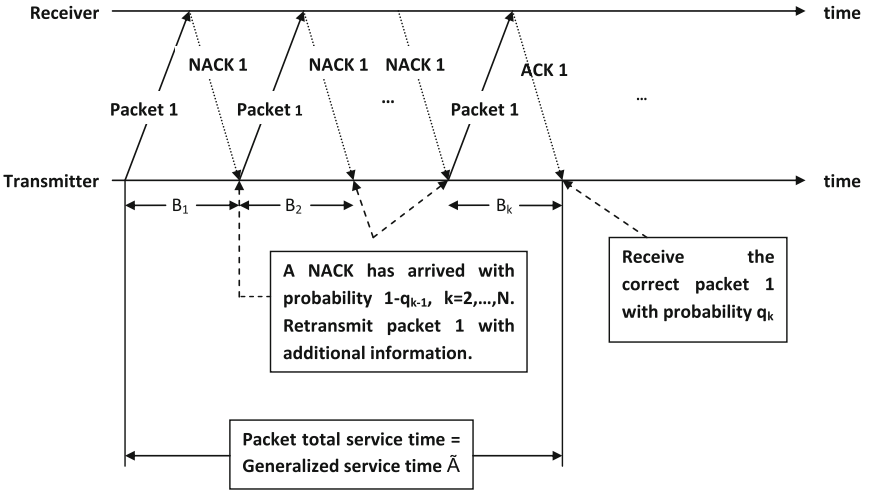


Fig. 1 Timing diagram of the HARQ process (First service cycle)

In the second service cycle the time to transmit the packet to the MS is B_0, $(B_0(x), b_0(x), \beta_0^*(s), \overline{b}_0, \overline{b}_0^{(2)})$. At the end of B_0, with probability q_0 the service is satisfactory, while with $\overline{q}_0$ an additional service time B_0 is requested. The reception of the packet by the MS is completed when a positive response is received by the BS.

When the buffer empties, the MS enables an inactivity timer, say I, which is arbitrarily distributed $(I(x), i(x), i^*(s), \overline{i}, \overline{i}^{(2)})$ and remains idle. If a packet arrives during I, the timer stops and the BS transmits the packet (see Fig. 2).

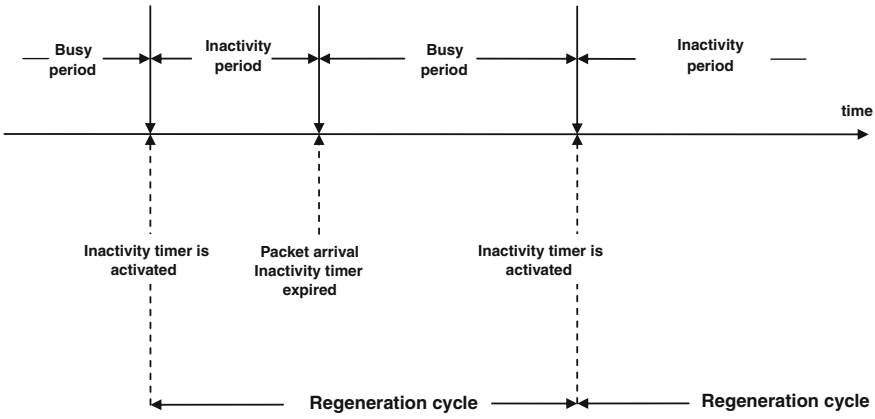


Fig. 2 Timing diagram I

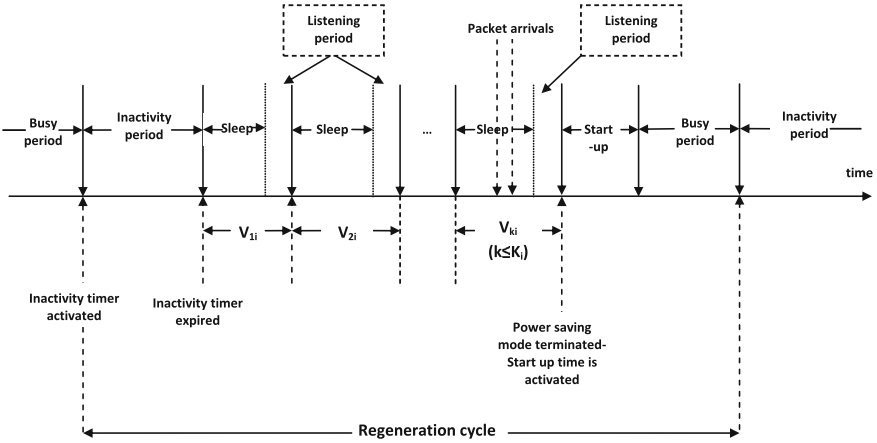


Fig. 3 Timing diagram II

If the inactivity timer expires, the MS switches to the sleep mode by selecting the initial vacation period which is arbitrarily distributed. The initial vacation period is chosen from the following set $\{V_{11}, V_{12}, \ldots, V_{1M}\}$, and with probability p_i, $i = 1, \ldots, M$, the MS selects the initial vacation period V_{1i}, where $\sum_{i=1}^{M} p_i = 1$. The selection of the initial vacation period affects the vacation policy. Thus, inactivity timer termination instants, are instants of actions that are taken from the MS to which sleep policy will be assigned. The probabilities p_i, $i = 1, \ldots, M$, can interpreted as the expected proportion of times the sleep policy i is selected (see also [10]). If $V_{ki}, k \geq 1$ ($V_{ki}(x)$, $v_{ki}(x)$, $v^*_{ki}(s)$, $\overline{v}_{ki}$, $\overline{v}^{(2)}_{ki}$), expires without packet arrivals, then the MS enters next vacation period, whose length depends on the number of preceding vacation periods (Fig. 3). If the number of consecutive vacation periods reaches up to a maximum value K_i, then the vacation period is fixed to $V_{K_i i}$, that is $V_{ki} \sim V_{K_i i}$, $k \geq K_i$, $i = 1, \ldots, M$. Each vacation period, consists of a sleep period, during which it powers down its physical components to save power, and a listening period of fixed length τ_i, $i = 1, \ldots, M$, during which it wakes up to check whether there are pending packets. These subperiods alternate until the MS is notified for buffered packets during the listening period.

In such a case, upon the termination of the listening period (see Fig. 3) the MS enables a start-up S (in order to powered on its components), which is arbitrarily distributed ($S(x)$,$s(x)$,$s^*(s)$,$\overline{s}$,$\overline{s}^{(2)}$). Upon expiration of S, the BS transmits the buffered packets to the MS.

Before proceeding with the analysis, we have to note that in case we neglect the principle of start-up periods, assume that $V_{ki} \backsim V$ (that is, the vacation intervals are independent and identically distributed random variables) and that the inactivity period cannot be preempted by packet arrivals, our model becomes a special case of a queueing model with a server of walking type, introduced in [9].

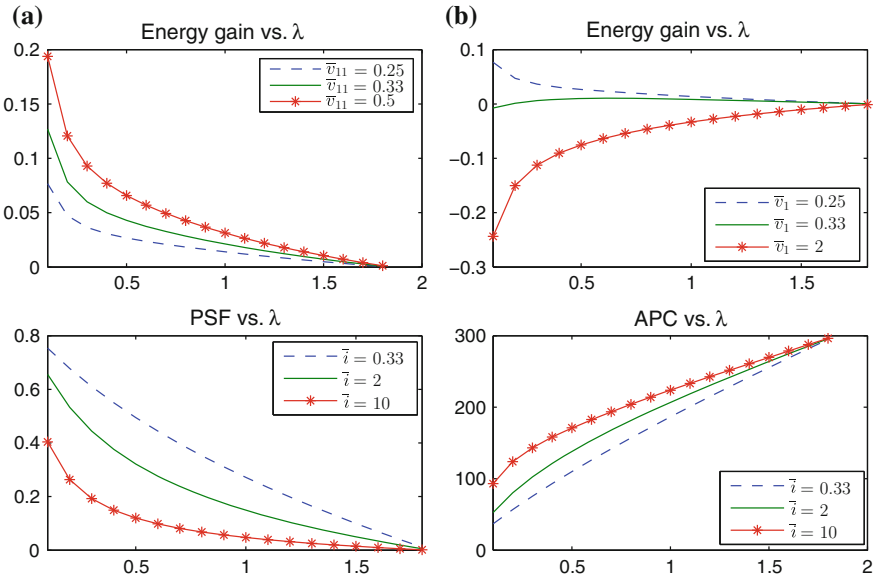


Fig. 4 Energy metrics

3 General Results

We now focus on the derivation of some crucial preliminary results that are essential for the following analysis. Define the generalized service time, say $\widehat{A}$, as the time elapsed from the epoch the BS starts the transmission of a packet until the epoch the MS successfully receives it. Denote also by $M(\widehat{A})$ the number of packets that arrive during $\widehat{A}$, and $a_j(t)\mathrm{dt} = P(t < \widehat{A} \leq t + \mathrm{dt}, M(\widehat{A}) = j)$, $a^*(z,s) = \sum_{j=0}^{\infty} \int_0^{\infty} e^{-st} a_j(t) z^j \mathrm{dt}$. If $e_j(t) = e^{-\lambda t} \frac{(\lambda t)^j}{j!}$, then

$$
\begin{aligned}
a_j(t) = {} & e_j(t) b_1(t) q_1 + \sum_{m_1=0}^{j} e_{m_1}(t) b_1(t) \overline{q}_1 * e_{j-m_1}(t) b_2(t) q_2 + \cdots \\
& + \sum_{m_1=0}^{j} e_{m_1}(t) b_1(t) \overline{q}_1 * \sum_{m_2=0}^{j-m_1} e_{m_2}(t) b_2(t) \overline{q}_2 * \ldots * \sum_{m_{N-1}=0}^{j-\sum_{i=1}^{N-2} m_i} e_{m_{N-1}}(t) \qquad (1) \\
& \times b_{N-1}(t) \overline{q}_{N-1} * e_{j-\sum_{i=1}^{N-1} m_i}(t) b_N(t) q_N + \sum_{m_1=0}^{j} e_{m_1}(t) b_1(t) \overline{q}_1 \\
& * \sum_{m_2=0}^{j-m_1} e_{m_2}(t) b_2(t) \overline{q}_2 * \ldots * \sum_{m_N=0}^{j-\sum_{i=1}^{N-1} m_i} e_{m_N}(t) b_N(t) \overline{q}_N * c_{j-\sum_{i=1}^{N} m_i}(t),
\end{aligned}
$$

where "*" means convolution and $c_k(t)$ is the pdf of the number of customers that arrive during the second service cycle. Writing an expression similar to (1) we can obtain,

$$a^*(z,s) = \sum_{i=1}^{N} \beta_i^*(s+\lambda-\lambda z) q_i \prod_{j=1}^{i-1} \overline{q}_j \beta_j^*(s+\lambda-\lambda z)$$
$$+ c^*(z,s) \prod_{i=1}^{N} \overline{q}_i \beta_i^*(s+\lambda-\lambda z),$$
$$c^*(z,s) = \sum_{j=0}^{\infty} \int_0^{\infty} e^{-st} c_j(t) z^j \mathrm{dt} = \frac{\mathrm{q}_0 \beta_0^*(\mathrm{s}+\lambda-\lambda \mathrm{z})}{1-(1-\mathrm{q}_0)\beta_0^*(\mathrm{s}+\lambda-\lambda \mathrm{z})}.$$

Define also $\rho = \sum_{i=1}^{N} \rho_i \overline{q}_{i-1} + \rho_0 \prod_{j=1}^{N} \overline{q}_j$, $\rho_i = \lambda \overline{b}_i$, $i = 1, \ldots, N$, $\rho_0 = \lambda \overline{b}_0 / q_0$. The next theorem investigates the number of zeros of the equation $z - a^*(z,s) = 0$. Its proof is a simple generalization of Takacs theorem [12].

Theorem 1 *For (i) $Re(s) > 0$, or (ii) $Re(s) \geq 0$, and $\rho > 1$ the equation $z - a^*(z,s) = 0$, has one and only one root, say $z = x(s)$, inside the region $|z| < 1$. Specifically for $s = 0$, $x(0)$ is its smallest positive real root with $x(0) < 1$ if $\rho > 1$ and $x(0) = 1$ for $\rho \leq 1$.* □

Define by $\widehat{V}_i$ the time elapsed from the epoch the MS enters the first vacation period V_{1i}, $i = 1, \ldots, M$ until the epoch it is ready to receive the first packet. Note that $\widehat{V}_i = \Theta_i + S$, where Θ_i is the so called effective vacation period. Moreover, in order for the effective vacation period to be terminated, at least one packet must arrive during Θ_i. Denote also by $N(\widehat{V}_i)$ the number of packets that arrive during $\widehat{V}_i$. If $\widehat{v}_j^{(i)}(t)\mathrm{dt} = P(t < \widehat{V}_i \leq t + \mathrm{dt}, N(\widehat{V}_i) = j)$ and $\widehat{v}_i^*(z,s) = \sum_{j=1}^{\infty} \int_0^{\infty} e^{-st} \widehat{v}_j^{(i)}(t) z^j$ dt then writing a similar expression as in (1) (see also [6]) we can finally obtain

$$\widehat{v}_i^*(z,s) = s^*(s+\lambda-\lambda z) \left[\prod_{m=1}^{K_i} v_{mi}^*(s+\lambda) \frac{v_{K_i i}^*(s+\lambda-\lambda z) - v_{K_i i}^*(s+\lambda)}{1 - v_{K_i i}^*(s+\lambda)} \right.$$
$$\left. + \sum_{k=1}^{K_i} \prod_{m=1}^{k-1} v_{mi}^*(s+\lambda)(v_{ki}^*(s+\lambda-\lambda z) - v_{ki}^*(s+\lambda)) \right]$$
$$= s^*(s+\lambda-\lambda z)\omega_i^*(s,z),$$

where $\omega_i^*(s,z), i = 1, \ldots, M$ (the term in brackets) is the LST of the probability generating function (pgf) of the number of packets that arrive during the effective vacation period of type i. Define by $\mathbf{H}^{(i)}$ the duration of the busy period, initiated by i packets. The busy period is the time elapsed from the epoch a service is initiated until the epoch the MS enables the inactivity period. If $g^{(i)}(t)\mathrm{dt} = \mathrm{P}(\mathrm{t} < \mathbf{H}^{(\mathrm{i})} \leq \mathrm{t} + \mathrm{dt})$, following the lines in Takacs [12] (pp. 60–63) we obtain $g^{(i)}(s) = \int_0^{\infty} e^{-st} g^{(i)}(t)\mathrm{dt} = \mathrm{x}^{\mathrm{i}}(\mathrm{s})$, $E(\mathbf{H}^{(1)}) = \rho/\lambda(1-\rho)$, where $x(s)$ is defined in Theorem 1.

Define by $\widetilde{V}_i$ the time elapsed from the epoch the MS enters the first vacation V_{1i}, until the epoch the MS enables the inactivity period, and $\widetilde{v}_i(t)\mathrm{dt} = P(t < \widetilde{V}_i \leq t + \mathrm{dt})$. By writing down a similar expression as in (1), we conclude that $\widetilde{v}_i^*(s) = \int_0^{\infty} e^{-st} \widetilde{v}_i(t)\mathrm{dt} = \widehat{v}_i^*(x(s), s)$, $i = 1, \ldots, M$.

Define now the regeneration cycle, say W, as the time elapsed from the epoch the server enables the inactivity timer until the epoch the next inactivity timer is about

to begin. The concept of regeneration cycle is necessary in order to obtain in the sequel the probabilities of server's state (MS's receiver state). Let $w(t)\mathrm{d}t = P(t < W \leq t + \mathrm{d}t)$, $w^*(s) = \int_0^\infty e^{-st} w(t)\mathrm{d}t$. Then, by writing an expression similar to (1) we can finally obtain, $w^*(s) = \lambda x(s)\frac{1-i^*(\lambda+s)}{\lambda+s} + i^*(s+\lambda)\sum_{i=1}^{M} p_i \widehat{v}_i^*(x(s), s)$. The expected duration of the above quantities are given for $i = 1, \ldots, M$,

$$E(\widetilde{V}_i) = \frac{1}{1-\rho}\left[\overline{s} + \sum\nolimits_{m=1}^{K_i-1} \overline{v}_{mi} \prod\nolimits_{j=1}^{m-1} v_{j-1i}^*(\lambda) + \frac{\overline{v}_{K_i i}}{1 - v_{K_i i}^*(\lambda)} \prod\nolimits_{m=1}^{K_i-1} v_{mi}^*(\lambda)\right], \tag{2}$$

$$E(W) = \frac{(1 - i^*(\lambda))}{\lambda(1-\rho)} + i^*(\lambda) \sum\nolimits_{i=1}^{M} p_i E(\widetilde{V}_i).$$

4 The Embedded Markov Chain

Let T_n be the epoch at which the nth packet is successfully delivered to the MS and let also X_n be the queue length just after T_n. Then,

$$X_{n+1} = \begin{cases} X_n - 1 + A_{n+1}, & \text{if } X_n > 0 \\ A_{n+1}, & \text{if } X_n = 0,\ Q_n > 0 \\ \Theta_{ni} + S_{n+1} + A_{n+1} - 1, & \text{if } X_n = 0,\ Q_n = 0, \text{ with prob.} p_i,\ i = 1, \ldots, M, \end{cases}$$

where A_n, S_n, Θ_{ni}, Q_n are the number of packets that arrive during the nth generalized service time, the start-up period for the transmission of the nth packet, the effective vacation period of type i, and the inactivity period following the nth packet departure instant. Definitely $\{X_n,\ n = 0, 1, \ldots\}$ constitutes a Markov chain, and define also the limiting probabilities $\pi_k = P(X = k)$, $q_k = P(Q = k)$, $a_k = P(A = k)$, $s_k = P(S = k)$, $k = 0, 1, \ldots$, $\omega_{ik} = P(\Theta_i = k)$, $k = 1, 2, \ldots$ and their corresponding pgfs for $|z| \leq 1$, $\Pi(z) = \sum_{k=0}^{\infty} \pi_k z^k$, $A(z) = \sum_{k=0}^{\infty} a_k z^k$, $\Omega_n(z) = \sum_{k=1}^{\infty} \omega_{nk} z^k$, $S(z) = \sum_{k=0}^{\infty} s_k z^k$.

Writing down the balance equations and employing the generating function analysis we conclude in

$$(z - A(z))\Pi(z) = \pi_0 A(z)[z(1 - q_0) + q_0 \sum\nolimits_{n=1}^{M} p_n \Omega_n(z) S(z) - 1], \tag{3}$$

where $A(z) = a^*(z, 0)$, $S(z) = s^*(\lambda - \lambda z)$, $\Omega_n(z) = \omega_n^*(z, 0)$, $\Omega(z) = \sum_{n=1}^{M} p_n \Omega_n(z)$ and $q_0 = \int_0^\infty e^{-\lambda t} i(t)\mathrm{d}t = \mathrm{i}^*(\lambda)$. Note that using Theorem 1, for $\rho < 1$, equation $z - A(z) = 0$ never vanishes inside the unit disk and thus we can devide and obtain $\Pi(z)$. Note also that $\Omega(z)$ is the pgf of the number of packets that arrive during the overall mixed vacation scheme. Since $\Pi(1) = 1$ we obtain $\pi_0 = \frac{1-\rho}{1+i^*(\lambda)[E(N(\Omega))+\lambda\overline{s}-1]}$ where $E(N(\Omega)) = \lambda \sum_{n=1}^{M} p_n [\sum_{i=1}^{K_n-1} \overline{v}_{in} \prod_{j=1}^{i-1} v_{j-1n}^*(\lambda) +$

$\frac{\overline{v}_{K_n n}}{1-v^*_{K_n n}(\lambda)} \prod_{i=1}^{K_n-1} v^*_{in}(\lambda)]$, is the mean number of packets that arrive during the overall mixed vacation scheme.

5 Performance Measures

In the following we derive useful performance and energy measures. The mean number of packets in the system in steady state is given by

$$E(X) = \rho + \frac{\rho^{(2)}}{2(1-\rho)} + \frac{\lambda^2 \overline{s}^{(2)} + 2\lambda\overline{s}(1-i^*(\lambda)) + i^*(\lambda)(\widehat{\Omega} + 2\lambda\overline{s}E(N(\Omega)))}{2(1+\lambda\overline{s}+i^*(\lambda)(E(N(\Omega))-1))}, \tag{4}$$

where $\widehat{\Omega} = \lambda^2 \sum_{n=1}^{M} p_n [\sum_{i=1}^{K_n-1} \overline{v}^{(2)}_{in} \prod_{j=1}^{i-1} v^*_{j-1n}(\lambda) + \frac{\overline{v}^{(2)}_{K_n n}}{1-v^*_{K_n n}(\lambda)} \prod_{i=1}^{K_n-1} v^*_{in}(\lambda)]$, and

$$\begin{aligned}\rho^{(2)} =& \lambda^2 \sum_{i=1}^{N} \overline{b}^{(2)}_i \prod_{j=1}^{i-1} \overline{q}_j + \frac{\lambda^2 \overline{b}^{(2)}_0}{q_0} \prod_{j=1}^{N} \overline{q}_j \\ &+ 2\sum_{i=1}^{N-1} \rho_i \sum_{m=i+1}^{N} \rho_m \prod_{j=1}^{m-1} \overline{q}_j + 2\rho_0 \prod_{j=1}^{N} \overline{q}_j (q_0\rho_0 + \sum_{i=1}^{N} \rho_i).\end{aligned}$$

Furthermore, using the Little's law we can easily calculate the mean waiting time $E(D) = E(X)/\lambda$. Clearly, the design and the development of energy-aware wireless systems require not only such average performance measures but also higher moments of packet queue length and transmission delay (or their coefficient of variation), which can be computed by appropriate differentiation of the generating function (3) and can provide useful information about their distribution. For reasons of space economy we omit them. Note also that these average measures can also be computed by the mean value analysis [3].

The MS's receiver activities are characterized by the regenerative process [11], which consists of the inactivity period, the start-up period, the modified vacation period and the busy period. In order to obtain the probabilities of MS's receiver state we have to evaluate the mean residence times of the receiver at these time periods during the regenerative cycle. Let $\widehat{I}$ be the time spent by the MS in the inactivity period. If $\widehat{i}(t)\mathrm{dt} = P(t < \widehat{I} \le t + \mathrm{dt})$, then $\widehat{i}(t) = e^{-\lambda t} i(t) + \lambda e^{-\lambda t}(1 - I(t))$. Thus, its LST is given by $\widehat{i}^*(s) = \lambda \frac{1-i^*(s+\lambda)}{s+\lambda} + i^*(s+\lambda)$ and $E(\widehat{I}) = \frac{1-i^*(\lambda)}{\lambda}$. Let $\widehat{S}$ the residence time of the MS's receiver in start-up during a regenerative cycle. Provided that the inactivity period has expired, $E(\widehat{S}) = i^*(\lambda)\overline{s}$.

The time period spent by the MS in the effective vacation period of type n during a cycle equals Θ_n and its LST is $\omega^*_n(s) = \omega^*_n(s, 1)$. The mean residence time in the effective vacation period of type n during a cycle is given by $E(\Omega_n) = i^*(\lambda)[\sum_{i=1}^{K_n-1} \overline{v}_{in} \prod_{j=1}^{i-1} v^*_{j-1n}(\lambda) + \frac{\overline{v}_{K_n n}}{1-v^*_{K_n n}(\lambda)} \prod_{i=1}^{K_n-1} v^*_{in}(\lambda)]$, where

$\prod_{l=n}^{m} h_l = 1$ for $m < n$. Therefore the mean residence time in the mixed vacation scheme during a cycle is $E(\Omega) = \sum_{n=1}^{M} p_n E(\Omega_n)$. Clearly each V_{jn} consists of a sleep and a listening period, which is fixed and equals τ_n. In order to obtain the mean residence time the MS stays in this state during a regeneration cycle, say $E(\zeta)$ we have to calculate the mean number of vacations of type n during a cycle, say $E(\zeta_n)$. Then, $E(\zeta) = \sum_{n=1}^{M} p_n E(\zeta_n)\tau_n$ where

$$\zeta_n = \begin{cases} i = 1, \ldots, K_n, \text{ with probability } (1 - v^*_{in}(\lambda)) \prod_{j=1}^{i-1}(v^*_{jn}(\lambda)), \\ i \geq K_n + 1, \quad \text{with probability } (1 - v^*_{K_n n}(\lambda))(v^*_{K_n n}(\lambda))^{i-1-K_n} \prod_{j=1}^{K_n}(v^*_{jn}(\lambda)). \end{cases}$$

$$E(\zeta_n) = i^*(\lambda)[\sum_{i=1}^{K_n-1} \prod_{j=1}^{i-1} v^*_{j-1n}(\lambda) + \frac{1}{1 - v^*_{K_n n}(\lambda)} \prod_{i=1}^{K_n-1} v^*_{in}(\lambda)],$$

Clearly, if a packet arrives during the inactivity period (with probability $1-i^*(\lambda)$), the timer stops and a busy period is initiated. Then, the mean duration of the residence time in a busy period during a cycle equals $E(B_{\widehat{I}}) = (1-i^*(\lambda))E(\mathbf{H}^{(1)}) = \frac{(1-i^*(\lambda))\rho}{\lambda(1-\rho)}$. Assume now that the inactivity period has expired, and the MS enters the sleep mode (with probability $i^*(\lambda)$). In such a case, the MS will remain in a busy state for a time period that is the sum of generalized service times of the number of packets that arrive during the mixed vacation period and the start-up time that is necessary for the first service. Using Wald's theorem, $E(B_{\widetilde{V}}) = i^*(\lambda)(\lambda\overline{s} + E(N(\Omega))E(\mathbf{H}^{(1)})$. Therefore, the total residence time the MS stays in a busy state during a cycle is $E(B) = E(B_{\widehat{I}}) + E(B_{\widetilde{V}}) = \rho E(W)$.

Clearly, the MSs receiver can be in the following states: in the inactive state (IN), in the start-up state (ST), in the sleep state of type n, $n = 1, \ldots, M$ (SL_n), in the listening state of type n (L_n) and in the busy state (B). Following Ross [11] the probabilities of MS's state are given by

$$P(B) = \rho, \qquad P(SL_n) = \frac{p_n(E(\Omega_n) - E(\zeta_n)\tau_n)}{E(W)}, \; n = 1, \ldots, M,$$

$$P(IN) = \frac{E(\widehat{I})}{E(W)}, \; P(L_n) = \frac{p_n E(\zeta_n)\tau_n}{E(W)}, \; n = 1, \ldots, M, \quad P(ST) = \frac{i^*(\lambda)\overline{s}}{E(W)}. \quad (5)$$

The Power Saving Factor $PSF = \frac{E(\Omega)-E(\zeta)}{E(W)}$ gives the percentage of time during which the MS is turned off and as a consequence does not consume power. Denote also by $C_B, C_{SL_n}, C_{L_n}, n = 1, \ldots, M, C_{\text{IN}}, C_{\text{ST}}$ the power consumption of the MS receiver in the busy state, in the sleep state of type n, in the listening state of type n, in the inactive, and in the start-up state respectively. Therefore, the average power consumption (APC) of the MS is obtained by

$$APC = \rho C_B + P(IN)C_{IN} + \sum_{n=1}^{M} P(L_n)C_{L_n} + P(ST)C_{ST} + \sum_{n=1}^{M} P(SL_n)C_{SL_n}. \quad (6)$$

Let us discuss a way about how we can use this model in real devices. When inactivity period is terminated, the following optimization problem takes place: Given the mean packet arrival rate, the mean duration of transmission times and an available set of sleep policies, find optimal values of p_is, $i = 1, \ldots, M$, $\sum_{i=1}^{M} p_i = 1$, $p_i \geq 0$ that minimize APC, asking $E(D) \leq \delta$, where δ is the maximum delay constraint (see also [10]). These values provide an optimal sleep mode selection policy.

Denote by $U_e = \frac{APC^* - APC}{APC^*}$, $U_d = \frac{E(D) - E(D^*)}{E(D)}$, the gain in energy and the increase in delay due to the employment of our mixed sleep-mode mechanism compared with other sleep-mode mechanisms and $\phi_p = U_e - pU_d$, where APC^*, $E(D^*)$, are the average power consumption and the mean waiting time for other sleep-mode mechanisms, that we compare with the proposed one. ϕ_p [7] is a metric to be used for comparing delay and energy consumption between our mixed vacation policy and an arbitrary other vacation policy. p is a delay penalty, which reflects the increase in delay because of the implementation of our mixed vacation policy, and must be set relatively high for delay sensitive applications and low when the primary focus is to extend the battery lifetime. Clearly, our policy is efficient to be used when $\phi_p > 0$.

6 Numerical Results

Assume now that all probability distributions are exponential and give the following default values: $N = 2$, $K_1 = 3$, $K_2 = 4$, $K_3 = 3$, $\overline{b}_1 = 0.33$, $\overline{b}_2 = 0.2$, $\overline{b}_0 = 25$, $\overline{q}_1 = 0.3$, $\overline{q}_2 = 0.1$, $\overline{q}_0 = 0.05$, $\overline{s} = 0.2$, $C_b = 300$ mW, $C_i = 150$ mW, $C_{l_1} = 90$ mW, $C_{l_2} = 100$ mW, $C_{l_3} = 80$ mW, $C_s = 130$ mW, $\tau_1 = 0.03$ s, $\tau_2 = 0.05$ s, $\tau_3 = 0.02$ s. The MS chooses among $M = 3$ sleep policies and with probability $p_1 = 0.5$ adopts the PSC I in IEEE 802.16e such that $V_{k1} \sim 2^{k-1} V_{11}$, $k = 2, 3$ and $V_{k1} \sim 2^{3-1} V_{11}$ for $k > 3$ with $\overline{v}_{11} = 0.25$. With probability $p_2 = 0.3$ adopts the DRX in LTE saving mode such that $V_{k2} \sim V_0$, $k = 1, 2, 3$ and $V_{k2} \sim V_1$, $k \geq 4$ with $\overline{v}_0 = 0.2$, $\overline{v}_1 = 0.25$, and with probability $p_3 = 0.2$ adopts the PSC II of IEEE 802.16e such that $V_{k3} \sim V_3$, $k \geq 1$, with $\overline{v}_3 = 0.33$.

Figure 4 provides information about the Energy gain, the *PSF*, and the *APC* for increasing values of λ. In Figures "Energy gain vs. λ (a,b)", we provide a comparison on energy gain between our mixed policy with $(p_1, p_2, p_3) = (0.5, 0.3, 0.2)$ and the DRX in LTE save mode $((p_1, p_2, p_3) = (0, 1, 0))$ for increasing values of λ. In the left-hand figure, we can observe tha our mixed policy is always better than the DRX mode. In the right-hand figure, when $\overline{v}_1$ (the mean duration of a long DRX cycle) is large enough, it is better to adopt the DRX in LTE as λ takes small values. On the other hand when $\overline{v}_1$ becomes small, our mixed sleep policy can guarantee better power improvements. As expected in Fig. 4 "*APC* versus λ" ("*PSF* vs. λ"), the more we increase λ, the more the *APC* increases (*PSF* decreases). The increase in *APC* (decrease in *PSF*) becomes more apparent when $\overline{i}$ increases.

Moreover, the analytical results has been validated against simulations (see [13] for more details). We focus on $PSF = T_s / T_0$, where T_0 is the total observation period

Table 1 Comparison of the analytical and simulation results

λ	0.1	0.15	0.2	0.25	0.3	0.35
PSF (Analytical)	0.793	0.788	0.784	0.775	0.7438	0.7315
PSF (Simulation)	0.787	0.7822	0.7814	0.7711	0.7399	0.7253
Error	0.076 %	0.0741 %	0.03327 %	0.051 %	0.0527 %	0.0855 %

in the simulation run, and T_s is the total sleep period within T_0. Table 1 indicates a very good match between the analytical and simulation results.

7 Conclusion

We mathematically investigated the power saving in modern wireless systems using a general queueing model with a probabilistic inhomogeneous vacation scheme, and a modified service time. The vacation scheme is general enough to describe the sleep policies that are applied in modern wireless standards, and is able to adopt a mixed sleep policy that may result in substantial improvements in power usage. Moreover, the service process properly describes the operation of the incremental redundancy retransmission scheme used in modern wireless systems.

Acknowledgments This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme and has received funding from the EU 7th Framework Programme (FP7/2007-2013) under grant agreement n. 246016. A part of this work was financially supported by a postdoctoral scholarship program financed by the Research Committee of the Aristotle University of Thessaloniki.

References

1. 3GPP TS 36.321 E-UTRA Medium Access Control (MAC) Protocol Specification. Rel. 8 v. 8.8.0 (2009)
2. IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. IEEE Std 802.16e- 2005 and IEEE Std 802.16-2004/Cor 1–2005 (Amendment and Corrigendum to IEEE Std 802.16-2004), 2006
3. I. Adan, J.A.C. Resing: Lecture notes in queueing theory. Eindhoven institute of technology (2001)
4. S. Alouf, E. Altman, A.P. Azad: Analysis of an M/G/1 queue with repeated inhomogeneous vacations with application to IEEE 802.16e power saving mechanism. In 5th International Conference on the Quantitative Evaluation of SysTems, Saint-Malo, France (2008), pp. 27–36
5. S. Baek, B.D. Choi: Analysis of discontinuous reception (DRX) with both downlink and uplink packet arrivals in 3GPP LTE. In 6th International Conference on Queueing Theory and Network Applications, IEEE Press, New York (2011), pp. 8–16

6. I. Dimitriou, A modified vacation queueing model and its application on the discontinuous reception power saving mechanism in unreliable long term evolution networks. Perform. Eval. **77**, 37–56 (2014)
7. S. De Vuyst, K. De Turck, D. Fiems, S. Wittevrongel, H. Bruneel, Delay versus energy consumption of the IEEE 802. 16e sleep-mode mechanism. IEEE T. Wirel. Commun. **8**(11), 5383–5387 (2009)
8. B.T. Doshi, Queueing systems with vacations—A survey. Queueing Syst. **1**(1), 29–66 (1991)
9. E. Gelenbe, R. Iasnogorodski, A queue with server of walking type. Annales de l' Institut Henry Poincaré (B), Probabilités et Statistiques **16**(1), 63–73 (1980)
10. L. Kong, G. Wong, D. Tsang, Performance study and system optimization on sleep mode operation in IEEE 802.16e. IEEE T. Wirel. Commun. **8**(9), 4518–4528 (2009)
11. S. Ross, *Stochastic Processes* (Wiley, London, 1996)
12. L. Takacs, *Introduction To The Theory Of Queues* (Oxford Univ. Press, New York, 1962)
13. S.-R. Yang, Y.-B. Lin, Modeling UMTS discontinuous reception mechanism. IEEE T. Wirel. Commun. **4**(1), 312–319 (2005)

Stability Criterion of a General Multiserver Multiclass Queueing System

Evsey Morozov

Abstract We consider a FCFS multiclass, multiserver queueing system with class-dependent and server-dependent service times. We find stability criterion of such a system using the regenerative approach. The key idea is to consider the basic queue-size process in the saturated system. Then we use renewal theory and a characterization of the limiting remaining regeneration time to establish that the basic regenerative process is positive recurrent. First we prove a sufficient condition, and then show that the system is unstable when this condition is violated. Some generalizations of this system are discussed as well.

Keywords Multiclass multiserver system · Class-dependent and server-dependent service times · Stability criterion · Regenerative method

1 Introduction

Stability of queueing models is one of the most intensively studied topics closely connected with the rapidly increasing development of communication and computer systems. Stability analysis of new complicatedly configured computer networks is a hard and timely problem requiring a refined and laborious mathematical technique, especially for non-Markovian models. Stability analysis establishes the region of the predefined parameters where the stationary regime of the basic queueing processes exists. The literature devoted to stability analysis of queueing systems is vast, and instead of giving a survey, we mention a few papers describing the state-of-the-art in the area and covering fluid analysis [7], the method of Lyapunov functions [12], and the regenerative method [15]. Much less attention has been paid to the models with nonidentical servers (*heterogeneous* systems), including partially accessible systems (see, for instance, [3, 8, 9], where various aspects of the analysis and further references can be found). We remark that such systems are not monotone since the service

E. Morozov (✉)
Institute of Applied Mathematical Research, Karelian Research Center
and Petrozavodsk State University, Petrozavodsk, Russia
e-mail: emorozov@karelia.ru

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_24

times depend on the assigned servers (in contrast to systems with identical servers). Stability analysis plays a secondary role in the mentioned works. For instance, an instability condition of a multiserver heterogeneous single-class system is obtained in the paper [8] as a byproduct of heavy-traffic analysis. The papers directly devoted to stability analysis are [4, 6, 11], which are closely related to our research. The analysis in [6] covers two-server system with class-dependent and server-dependent service times, while for a larger number of servers it is assumed that only one type of dependence is allowed. The paper [4] extends results of [6] to some networks with feedback, while [11] analyses the stability of controlled queues with Poisson inputs. We will show that (under slightly different assumptions) the system we consider can be investigated by the fluid method developed in [6]. Unlike the mentioned works based on the fluid stability analysis, we apply the probabilistic regenerative approach to prove stability of a general multiclass, multiserver system with class-dependent and server-dependent service times. A key element in the approach used is a characterization of the limit behavior of the remaining renewal time in the process generated by regenerations. (For more detail see [15].) It is worth mentioning that, unlike fluid analysis, this approach covers the *boundary case* when the input rate equals the maximal capacity of the system. As a result, we find stability criterion for this system expressed in the terms of given parameters. Stability analysis of queueing systems has been an intriguing area of research in the past 20 years. We stress the importance of stability analysis in a practical context because an explicit form of stability criterion allows to chose minimal capacity of the system to guarantee, in particular, predefined quality of service requirement.

In summary, the main contribution of this work is a novel probabilistic proof of the stability criterion for a multiserver, multiclass system with class-dependent and server-dependent service times. This proof, unlike fluid analysis, covers the boundary case and allows some useful generalizations.

The paper is organized as follows. In Sect. 2 we describe the basic model. Section 3 contains the formulation and the proof of the main stability result. Section 4 is devoted to instability analysis, showing that the established condition is indeed a stability criterion. Finally, in Sect. 5, connections of the basic model with the known models and some generalizations are discussed.

2 Description of the Model

We consider an M-server multiclass queueing systems with renewal input with the (i.i.d.) interarrival times $\{\tau_n,\ n \geq 1\}$ with the rate $\lambda := 1/\mathsf{E}\tau \in (0,\ \infty)$ and with K classes of users. (Throughout the paper we omit the corresponding index to denote a generic element of an i.i.d sequence.) It is assumed that each new user belongs to class k with the probability p_k, and the input rate of class-k users is then $\lambda_k := \lambda p_k,\ k = 1, \ldots, K$. By FCFS service discipline, p_k is also the probability that an arbitrary user entering any server is class-k user as well. We introduce the independent sequences

of the i.i.d. service times $\{S_{ik}^{(n)},\ n \geq 1\}$ of class-k users at server i with (generic) service time S_{ik} with rate $\mu_{ik} = 1/\mathsf{E}S_{ik} \in (0, \infty)$, $i = 1, \ldots, M$. Assume that $I_{ik}^{(n)} = 1$, if the nth user entering server i is class-k one, and $I_{ik}^{(n)} = 0$, otherwise (index n counts users intended for server i only), $n \geq 1$. It then follows that

$$S_i(n) := \sum_{k=1}^{K} I_{ik}^{(n)} S_{ik}^{(n)} \tag{1}$$

is the service time of the nth user at server i. Since, for each k, the i.i.d. sequences $\{I_{ik}^{(n)},\ n \geq 1\}$ are stochastically equivalent for all i, then in particular, $\mathsf{E}I_{ik} = p_k$ for all k. Note that, for each i, the variables $\{S_i(n),\ n \geq 1\}$ are i.i.d. with a generic element S_i and the expectation

$$\mathsf{E}S_i = \sum_{k=1}^{K} \frac{p_k}{\mu_{ik}},\quad i = 1, \ldots, M. \tag{2}$$

We assume that each free server can be selected for new service with a probability, which is lower bounded by a constant $d > 0$. Let $\nu_i(t)$ be the number of the users (in the system) at instant t which are intended for server i, and let $\nu(t) = \sum_{i=1}^{M} \nu_i(t)$. Also denote by $\{t_n,\ n \geq 1\}$ the arrival instants. Then regeneration epochs $\{T_n\}$ of the processes $\{\nu(t)\}$ (and other processes describing the system) occur when a user meets an empty system, and are defined recursively by the following standard way

$$T_{n+1} = \inf_k (t_k > T_n : \nu(t_k^-) = 0), \quad n \geq 0\ (T_0 := 0,\ \inf \emptyset := \infty). \tag{3}$$

Remark that at each such an instant the new regeneration cycle is initiated by a class-k user with the probability p_k. Let T be a generic regeneration period under zero initial state, that is, $T = \inf(t_k > 0 : \nu_k = 0 | \nu_1 = 0)$ provided $t_1 = 0$. The process $\{\nu(t)\}$ is called *positive recurrent* if [13]

$$\mathsf{E}T < \infty \ \text{ and } \ T_1 < \infty \ \text{ with probability 1 (w.p.1).} \tag{4}$$

If moreover T is *nonlattice* (for instance, if the input is Poisson), then the stationary distribution of $\nu(t)$ (and other related processes) exists as $t \to \infty$ [1]. (We call it *stationary regime*.)

3 Stability Analysis

In this section, we formulate and prove the following main stability result. Denote $\mu_i = 1/\mathsf{E}S_i$, $\mu = \sum_{i=1}^{M} \mu_i$, see (2).

Theorem 1 *Assume that*

$$\lambda < \mu \tag{5}$$

and

$$\max_{i,k} \mathsf{P}(\tau > S_{ik}) > 0. \tag{6}$$

Then the regenerative queue-size process $\{\nu(t),\ t \geq 0\}$ *is positive recurrent under arbitrary fixed initial state* $\nu(0)$.

Proof Denote by $A_k(t)$ the number of class-k users arrived in the time interval $(0,\ t]$, and let $A(t) := \sum_k A_k(t),\ k = 1, \ldots, K$. Also denote by $D_i(t)$ the number of departures from server i in $(0,\ t],\ i = 1, \ldots, M$. Finally, denote by $\hat{D}_i(t)$ the number of departures from server i in $(0,\ t]$ if it is busy all time $[0,\ t]$, and assume that $\hat{D}_i(0) = 1$. Thus, the family of the (zero-delayed) renewal processes $\{\hat{D}_i(t),\ i = 1, \ldots, M\}$ corresponds to the *saturated regime* of the system. It immediately follows from the renewal theory [1, 16] that both the convergence w.p.1

$$\lim_{t\to\infty} \frac{1}{t}\hat{D}_i(t) = \mu_i, \quad \lim_{t\to\infty} \frac{1}{t}\sum_{i=1}^{M} \hat{D}_i(t) = \mu, \quad \lim_{t\to\infty} \frac{1}{t}\sum_{k=1}^{K} A_k(t) = \lambda, \tag{7}$$

and the convergence in average

$$\lim_{t\to\infty} \frac{1}{t}\mathsf{E}\hat{D}_i(t) = \mu_i, \quad \lim_{t\to\infty} \frac{1}{t}\sum_{i=1}^{M} \mathsf{E}\hat{D}_i(t) = \mu, \quad \lim_{t\to\infty} \frac{1}{t}\sum_{k=1}^{K} \mathsf{E}A_k(t) = \lambda \tag{8}$$

hold. Now, for each server i, we use the same service times in the process $\hat{D}_i(t)$ and in the real departure process $D_i(t)$, implying $D_i(t) \leq \hat{D}_i(t)$ w.p.1 for all t. (Indeed we use a *coupling*, and w.p.1 inequality holds actually for stochastically equivalent processes [1].) Denote

$$D(t) = \sum_{i=1}^{M} D_i(t),\ A(t) = \sum_{k=1}^{K} A_k(t),\ \hat{D}(t) = \sum_{i=1}^{M} \hat{D}_i(t).$$

Because $A(t) \geq D(t)$, it then follows from (5), (7) that

$$\liminf_{t\to\infty} \frac{1}{t}(\hat{D}(t) - D(t)) \geq \mu - \lambda := \delta > 0. \tag{9}$$

Let

$$I(t) = \int_0^t \mathbf{I}(v(u) < M)du \tag{10}$$

be the total time in $[0,\ t]$ when at least one server is free. ($\mathbf{I}(A)$ denotes the indicator of the event A.) Also denote

$$I_i(t) = \int_0^t \mathbf{I}(v_i(u) = 0)du,\ \ i = 1, \ldots, M, \tag{11}$$

and remark that $I_i(t) \le I(t),\ i = 1, \ldots, M;\ t \ge 0$. Now, for each t and any server i, we "couple together" the actual service times (in the original departure process $D_i(t)$) which start in the interval $[0,\ t]$, then shift the obtained renewal sequence to the origin, and finally invoke the independence between $I_i(t)$ and the service times that start in interval $(t - I_i(t),\ t]$. Then the following upper bound holds:

$$\begin{aligned} \Delta_i(t) :&= \hat{D}_i(t) - D_i(t) \le \hat{D}_i(t) - \hat{D}_i(t - I_i(t)) + 1 \\ &\le_{st} \hat{D}_i(I_i(t)) + 1 \le \hat{D}_i(I(t)) + 1, \end{aligned} \tag{12}$$

where $\le_{st}$ means stochastic inequality. As a result, the difference $\Delta(t)$ between $\hat{D}(t)$ and the number of actual service completions $D(t)$ is upper bounded as:

$$\Delta(t) := \sum_{i=1}^{M} \Delta_i(t) = \hat{D}(t) - D(t) \le \hat{D}(I(t)) + M. \tag{13}$$

Thus, by (9),

$$\liminf_{t\to\infty} \frac{1}{t}\hat{D}(I(t)) \ge \liminf_{t\to\infty} \frac{1}{t}\Delta(t) \ge \delta. \tag{14}$$

Since $\hat{D}(t) < \infty$ w.p.1 for each t, then it follows from (14) that

$$I(t) \to \infty,\ \ t \to \infty \ \text{ w.p.1}. \tag{15}$$

Moreover, by Fatou's lemma,

$$\liminf_{t\to\infty} \frac{1}{t}\mathsf{E}\hat{D}(I(t)) \ge \delta. \tag{16}$$

By (8), for any $\varepsilon > 0$ there exists t_0 such that $\mathsf{E}\hat{D}(t) \le (\mu + \varepsilon)t$ for $t \ge t_0$. Thus, we have for $t \ge t_0$:

$$\mathsf{E}\hat{D}(I(t)) \le \mathsf{E}\hat{D}(t_0) + \int_{t_0}^{t} \mathsf{E}\hat{D}(u)\mathsf{P}(I(t) \in du) \le \mathsf{E}\hat{D}(t_0) + (\mu+\varepsilon)\mathsf{E}I(t). \quad (17)$$

Now we obtain from (16), (17) that

$$\liminf_{t\to\infty} \frac{1}{t}\mathsf{E}I(t) = \frac{1}{t}\liminf_{t\to\infty} \int_0^t \mathsf{P}(\nu(u) < M)du \ge \frac{\delta}{\mu+\varepsilon}, \quad (18)$$

implying

$$\limsup_{t\to\infty} \mathsf{P}(\nu(t) < M) > 0. \quad (19)$$

In other words, there exist $\delta_0 > 0$ and a deterministic sequence $z_n \to \infty$ such that

$$\inf_n \mathsf{P}(\nu(z_n) < M) \ge \delta_0. \quad (20)$$

Since, for each i, the sequence of the remaining service times $\{S_i(z_n),\ n \ge 1\}$ is tight [15], then the remaining workload sequence $\{W(z_n) := \sum_i S_i(z_n),\ n \ge 1\}$ is tight as well. In other words, there exists a constant $C < \infty$ such that

$$\inf_n \mathsf{P}(\nu(z_n) < M,\ W(z_n) \le C) \ge \frac{\delta_0}{2}. \quad (21)$$

By $\mathsf{E}\tau < \infty$ and condition (6), there exist server i_0, class k_0 and constants $\varepsilon_0 > 0,\ \epsilon > 0,\ C_0 < \infty$ such that

$$\mathsf{P}(C_0 \ge \tau > S_{i_0k_0} + \varepsilon_0) \ge \epsilon > 0. \quad (22)$$

Assume that $\hat{I}(k_0) = 1$ if an arriving user is class-k_0, and $\hat{I}(k_0) = 0$, otherwise. Now fix some z_n satisfying (20) and, on the event$\{\nu(z_n) < M,\ W(z_n) \le C\}$, realize $G := \lceil C/\varepsilon_0 \rceil$ independent events

$$\mathcal{E} := \{\hat{I}(k_0) = 1\} \cap \{C_0 \ge \tau > S_{i_0k_0} + \varepsilon_0\}$$

related to the users arriving since the instant z_n. As a result, we obtain that in the interval $[z_n,\ z_n + C_0G]$ a class-k_0 user arrives which meets at least server i_0 empty, and this happens with a probability which is lower bounded by $(p_{k_0}\epsilon)^G > 0$. Since this instant we will realize the events $\mathcal{E}$ and in addition route each new (class-k_0) user to an empty server i_0. (The latter is possible with a probability $\ge d$.) Thus, during each interarrival time (which is $\le C_0$, see (22)) the workload being in each server will decrease not less than by ε_0. Continuing in a similar way, we see that in a finite interval $[z_n,\ z_n + \mathcal{D}]$ a class-k_0 user arrives which meets the system completely

empty, and it happens with a positive probability γ. Moreover, both the length $\mathcal{D}$ of the interval and the probability γ are independent of n and the instant z_n. (This procedure is described in more detail in [13].) It then follows that the remaining regeneration time at instant z_n,

$$R(z_n) := \min_j (T_j - z_n : T_j - z_n > 0) \not\Rightarrow \infty, \ n \to \infty \ \text{(in probability)}.$$

Since the sequence $\{z_n\}$ is deterministic, then $R(t) \not\Rightarrow \infty$ as $t \to \infty$, and it follows that $\mathsf{E}T < \infty$ [5]. If, in addition, period T is nonlattice and the first regeneration period $T_1 < \infty$ w.p.1, then the stationary regime, that is the weak limit $\nu(t) \Rightarrow \nu$, exists [1, 16]. (It holds for other queueing processes as well.)

To prove that $T_1 < \infty$ w.p.1, we note that, by (15), the total time the process $\{\nu(t)\}$ spends in the compact set $\mathcal{M} := [0, M]$ is infinite w.p.1. On the other hand, we can show as in [15] that the total time the process spends in the set $\mathcal{M}$ during a *regeneration period* is finite w.p.1. It implies that $T_1 < \infty$ w.p.1, and thus the statement of Theorem 1 holds for an arbitrary fixed initial state $\nu(0)$. The proof is completed.

4 Instability

In this section, we study the behavior of the queue-size process $\{\nu(t)\}$ when stability condition (5) is violated. As we will show, in this case no stationary regime exists. We call system *strongly unstable* if $\nu(t) \to \infty$ w.p.1, and *weakly unstable*, if $\nu(t) \Rightarrow \infty$ as $t \to \infty$. The following instability result holds for the considered model.

Theorem 2 *Assume that condition*

$$\lambda \geq \mu \tag{23}$$

holds. Then
(i) if $\lambda > \mu$, then $\nu(t) \to \infty$ w.p.1 for an arbitrary fixed initial state $\nu(0)$;
(ii) if $\lambda = \mu$, $\nu(0) = 0$, and condition (6) holds, then $\nu(t) \Rightarrow \infty$;
(iii) if $\lambda = \mu$ and (6) holds, then no stationary distribution of $\nu(t)$ exists as $t \to \infty$ for any $\nu(0)$.

Proof To prove (i), we use a coupling to obtain inequality

$$\nu(t) = A(t) + \nu(0) - D(t) \geq A(t) + \nu(0) - \hat{D}(t),$$

implying

$$\liminf_{t\to\infty} \frac{\nu(t)}{t} \geq \lambda - \mu > 0. \tag{24}$$

To prove (ii), we remark that $\nu(0) = 0$ means that the first regeneration period T_1 has the same distribution as typical period T. We will use the proof by contradiction, and assume that $\mathsf{E}T < \infty$, implying $T_1 < \infty$ w.p.1. Then all processes describing the system are positive recurrent. Moreover, denote by $I_0(t)$ the total time, in interval $[0,\ t]$, when the system is fully empty, and denote by I_0 an empty period following a busy period B during one regeneration cycle. (Within a busy period B at least one server is occupied.) In other words, the following stochastic equality holds, $T =_{st} B + I_0$. By condition (22), a regeneration period may contain only class-k_0 user served by server i_0 and the following empty period is not less than ε. This happens with a probability $\geq p_{k_0} d\epsilon$. As a result, we obtain that

$$\mathsf{E}I_0 \geq p_{k_0}\, d\, \varepsilon\, \epsilon > 0. \tag{25}$$

By (25) and positive recurrence, $I_0(t) \to \infty$ w.p.1 as $t \to \infty$. Since $\nu(0) = 0$ then $A(t) = D(t) + \nu(t)$. Moreover it is easy to check that, for each i, $I_0(t) \leq I_i(t)$ (see (11)) and that $\hat{D}_i(t) - D_i(t) \geq \hat{D}_i(I_i(t)) - 1$. Then we obtain that

$$\hat{D}(t) - A(t) = \hat{D}(t) - D(t) - \nu(t) \geq_{st} \hat{D}(I_0(t)) - M - \nu(t),\ t \geq 0. \tag{26}$$

By positive recurrence, $\nu(t) = o(t),\ t \to \infty$. Then, dividing both sides of (26) by t and taking $t \to \infty$, we obtain from the regeneration theory [16] and from (25), (26) that

$$\mu - \lambda \geq \lim_{t\to\infty} \frac{\hat{D}(I_0(t))}{I_0(t)} \cdot \frac{I_0(t)}{t} = \mu \frac{\mathsf{E}I_0}{\mathsf{E}T} > 0. \tag{27}$$

This contradicts the assumption $\lambda = \mu$ implying $\mathsf{E}T = \infty$. Now we assume that $\nu(t) \not\Rightarrow \infty$. Then, applying the proof of Theorem 1 after formula (20), we obtain that $\mathsf{E}T < \infty$. This contradiction shows that indeed $\nu(t) \Rightarrow \infty$, and (ii) is proved.

It remains to consider (iii), in which case $T_1 \neq_{st} T$ in general. If $\mathsf{P}(T_1 < \infty) = 1$ then we have previous case. Otherwise, if $\mathsf{P}(T_1 = \infty) > 0$, then (because condition (6) holds) it is possible to show [15] that $\nu(t) \Rightarrow \infty$ on the event $\{T_1 = \infty\}$. Hence, there is no stationary distribution of $\nu(t)$ as $t \to \infty$, and the proof of (iii) is completed.

5 Concluding Remarks

Consider original M-server model in which service time S_{ik} of class-k user at server i depends on i only (*class-independent* service times). Thus, server i rate becomes

$$\mu_i := \frac{1}{\mathsf{E}S_{ik}},\ k = 1, \ldots, K,$$

and the system is heterogeneous single-class. Because $\sum_i p_i = 1$, then it follows that condition (5) remains formally the same:

$$\lambda \equiv \sum_{i=1}^{K} \lambda_i < \sum_{i=1}^{M} \mu_i.$$

Assume now that the service rate depends on class number only (*server-independent service times*), that is for each class-k user

$$\mu^{(k)} := \frac{1}{\mathsf{E}S_{ik}}, \ i = 1, \ldots, M.$$

Then we obtain a multiclass homogeneous system. Denote $\rho_k = \lambda_k/\mu^{(k)}$, then condition (5) becomes (after some algebra) $\sum_{k=1}^{K} \rho_k < M$, which is well-known for a single-server multiclass system.

Now we show how our basic model relates to the general model considered in [6], where only a two-server system is considered provided service time is server- and class-dependent. Moreover, it is shown in [6] and [4] that for more than two servers stability may depend on whole service time distributions, and not only on the first moments. First we note that our FCFS system is join-shortest-workload. Then, since each user entering each server i has (generic) service time S_i (see (1)), it follows that the system can be reduced to a *single-class* system with stability condition (5), see Theorem 2.5 in [6] (with assumption (6) replaced by the unboundedness of the interarrival times).

Now we mention some important generalizations of the model, which preserve regeneration property and can be analyzed by our approach.

Consider a discrete time system with K independent renewal processes. In this case, it is possible to construct regeneration instants of the merged flow with the rate $\lambda := \lim A(t)/t$, see [14]. Then one can show that, under appropriately modified condition (6), stability condition (5) remains the same, provided that the probabilities p_k are defined as $\lim_{t\to\infty} A_k(t)/t = p_k$.

Another generalization is obtained if the system is originally fed by a *regenerative* input, in which case some mild conditions on the structure of the input are required, see [13].

Finally we mention a modification of the basic model in which changes of positions of some awaiting users are allowed, but the limits $A_k(t)/t \to p_k$ remain unchanged for all k. In this case stability (5) holds true as well. The latter model relates to the so-called *discriminatory service discipline* keeping a trade-off between queues of different classes (for $K = 2$ classes, see [10]). Such a model is also closely related with the system with *relative priority* (see [2] and references therein). Another application of this model is the system with *multiple reservation* [17].

Acknowledgments The author cordially thanks S. Foss and N. Chernova for their useful comments and pointing out the papers [4, 6, 11]. The research is partially supported by the Program of

Strategy Development of Petrozavodsk State University in the framework of the research activity and the Polish project NCN nr 4796/B/T02/2011/40 *"Models for transmissions dynamics, congestion control and quality of service in Internet"*.

References

1. S. Asmussen, *Applied Probability and Queues* (Springer, New York, 2003)
2. U. Ayesta, A. Izagirre, I. M. Verloop, Heavy-trafc analysis of the discriminatory random-order-of-service discipline. ACM SIGMETRICS Performance Evaluation Review—Special Issue on IFIP PERFORMANCE 2011–29th international symposium on computer performance, modeling, measurement and evaluation, 39, 2, pp. 41–43 (2011)
3. O.J. Boxma, Q. Deng, A.P. Zwart, Waiting-Time asymptotics for the M/G/2 queue with heterogeneous servers. Queueing Syst. **40**(1), 5–31 (2002)
4. J. Dai, J. Hasenbein, B. Kim, Stability of join-the-shortest-queue networks. Queueing Syst. **57**, 129–145 (2007)
5. W. Feller, *An Introduction to Probability Theory and its Applications*. (Wiley, New York, 1971)
6. S. Foss, N. Chernova, On the stability of a partially accessible multi-station queue with state-dependent routing. Queueing Syst. **29**, 55–73 (1998)
7. S. Foss, T. Konstantopoulos, An overview on some stochastic stability methods. J. Oper. Res. Soc. Jpn. **47**(4), 275–303 (2004)
8. D.L. Iglehart, W. Whitt, Multiple channel queues in heavy traffic. I. Adv. Appl. Prob. **2**, 150–170 (1970)
9. J.H. Kim, H.-S. Ahn, R. Righter, Managing queues with heterogeneous servers. J. Appl. Probab. **48**(2), 295–595 (2011)
10. J. Kim, J. Kim, B. Kim, Analysis of the M/G/1 queue with discriminatory random order service policy. Perform. Eval. **68**(3), 256–270 (2011)
11. I.M. MacPhee, L.J. Muller, Stability criteria for controlled queueing systems. Queueing Syst. **52**, 215–229 (2006)
12. S. Meyn, D. Down, Stability of generalized Jackson networks. Ann. Appl. Probab. **4**, 124–148 (1994)
13. E. Morozov, Weak regeneration in modeling of queueing processes. Queueing Syst. **46**, 295–315 (2004)
14. E. Morozov, D. Fiems, Bruneel, H: Stability analysis of multiserver discrete-time queueing systems with renewal-type server interruptions. Perform. Eval. **68**, 1261–1275 (2011)
15. E. Morozov, R. Delgado, Stability analysis of regenerative queues. Autom. Remote control **70**(12), 1977–1991 (2009)
16. W.L. Smith, Regenerative stochastic processes. Proc. Roy. Soc. ser. A **232**, 6–31 (1955)
17. S. De Vuyst, S. Wittevrongel, D. Fiems, H. Bruneel, Controlling the delay trade-off between packet flows using multiple reserved places. Perform. Eval. **65**(6–7), 484–511 (2008)

Performance Analysis of Cluster-Based Web System Using the QPN Models

Tomasz Rak

Abstract The paper presents the cluster-based web system for that the time rate of changing a system offer is compared to the users' interaction time with a system. Systems with rapidly changing offer are used in different domains including the electronic trading to build scalable distributed systems. In the paper, such systems are modeled by queueing Petri nets. The aim of this work was to develop simulation models of Internet system by queueing Petri net modeling environment tool to performance analysis. Developed models allow to evaluate their performance (e.g., response time). In our experimental analysis, we use clustered web environment with Apache DayTrader benchmark as an e-trading system. It helped to determine the parameters for the simulation models and then to verify the simulation results. The paper includes the selected results of models simulation. Our approach predicts the performance of the given application deploy on a selected platform.

1 Introduction

Modeling and design of Internet systems (ISs) develop in two ways. On the one hand, formal models which can be used to analyze performance parameters are proposed: queueing nets (QNs) or colored Petri nets (CPNs) [1–3]. Sometimes elements of the control theory are used to manage the movement of packages in systems [4, 5]. Experiments are the second way [6]. Application of simulations and experiments greatly influences validity of the systems being developed. The convergence of simulation results with the real system results (experiments) confirms the correctness of the formal modeling methods.

T. Rak (✉)
The Faculty of Electrical and Computer Engineering,
Department of Computer and Control Engineering,
Rzeszow University of Technology, Rzeszow, Poland
e-mail: trak@kia.prz.edu.pl

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_25

First, we joined two formalisms. Our earlier works [2, 3, 7] are based on QN and timed colored Petri nets (TCPN). A distributed IS model, initially described in compliance with QN rules, is mapped onto TCPN structure by means of queueing system templates. We have used two types of formal models that have been exploited in the industry.

In this paper, we consider cluster-based web system (CWS). The CWS is interesting from the practical point of view. We created system models using queueing Petri nets (QPNs), which allow the performance engineering (PE) analysis. Then, we verified the constructed models with the real experimental environment as a benchmark. We considered cases in which the number of requests per second (RPS) is hundreds or thousands. Such situation may cause a rejection or time increase for a large number of requests, due to timeliness loss. Waiting requests increase the response time for all requests. We assume that a large part of requests may concern the system offer, that could be used by previous requests. Realization of transactions that are related to the system offer must take into account the results of previous transactions associated with this system offer [7]. A stock exchange system, where transactions are carried out online, could be their representative.

Energy consumption in information and communications technology is growing. Power consumption depends on the load and on the number of running nodes in CWS. It is therefore important to study ways to reduce energy consumption [8].

The remaining work is organized as follows. Section 2 presents CWS architecture. In Sect. 3, we shortly present used formal methods of ISs modeling. In Sect. 4, we introduce DayTrader benchmark and real CWS (hardware and software). This section presents experimental environment for formal models verification. Section 5 focuses on the results of queueing Petri net modeling environment (QPME) models simulation. The final section contains concluding remarks.

2 CWS Architecture

In the case of CWS performance modeling, an IS architecture must be specified. A typical IS architecture is made up of several layers containing: presentation server, application server (AS), database server (DBS), and data storage system.

The presented architecture in our approach and earlier [2] has been simplified to two layers: presentation and application of front-end layer (FE) and back-end layer (BE) which keeps the system data. Adaptation to the real experimental environment was the main reason for this simplification. Architecture composed of these layers is used for e-business systems [1]. The presented double-layer system architecture realizes the same CWS functions. Functionalities are merged in this architecture, as opposed to one previous solution [7]. Clustering mechanism is used in FE layer and allows to process the requests simultaneously. Similar solutions were used in [1] but with many simplifications and without exact IS parameters. The benchmark

used in this work has got realistic workload. The approach proposed in this paper may be treated as continuation and extension of previous solutions [2, 3, 7] with QPN formalism. The presented architecture components should be considered during solutions analysis.

3 Queueing Petri Nets

In our solution, we propose QPN formalism [9]. There is a very popular formal method in functional and performance modeling (performance analysis). These nets provide sufficient power of expression for modeling and analyzing complex online systems. The choice of QPN was caused by a possibility of obtaining information of different characters. The main idea of QPN is to add queueing and timing aspects to the net places.

Queuing nets (QNs)—quantitative analysis—have a queue and scheduling discipline and are suitable for modeling competition for equipment. Petri nets (PNs)—qualitative analysis—have tokens representing the tasks and are suitable for modeling software. QPNs have the advantages of QNs (e.g., evaluation of the performance of the system, the efficiency of the network) and PNs (e.g., logical assessment of the correctness of the system).

QPN consists of queueing places (resource or state) which consist of two components: the queue and depository for tokens which completed their service at the queue. Tokens, when fired into a queueing place by input transitions, are inserted into the queue according to the queue's scheduling strategy. After service, tokens are not available for output transactions. Tokens are immediately moved to the depository, where they become available for output transitions [1].

The response time was chosen to analyze from many PE parameters:

$$R = \sum_{n}^{i=1} R_i^{'}, \tag{1}$$

where n—number of queues, $R_i^{'} = Q_i + D_i$—residence time, $Q_i = \sum_n^{i=1} q_i$—queue time, $D_i = \sum_n^{i=1} d_i$—service demand. Total response time (Eq. 1) is a sum of all queues and depositories response times in simulation model without client queue response time (client think time).

4 Real System Environment

Here, we present the results of our experimental analysis. The first goal is to check the service demand parameters for FE and BE nodes. The second goal is to check the simulation results.

Deployment details are as follows: network segment (1 Gb/s) and 6 nodes (HP ProLiant DL180 G6). Software environment consists of 64-bit Linux operating systems, Clients Load Generator (CLG), Apache Tomcat Connector (load balancer), GlassFish 3.1 (as AS—first is Domain Administration Server), and Oracle 11g (as DBS). All important configuration parameters were described in tables before each test.

By client business transactions, we mean the operations: Login/Logout, Get Home, Get Portfolio, Show Quote, Buy Quote, Sell Quote, Show Account, and Update Profile. Each business transaction emulates a specific type of client session. The CLG is implemented using multi-threaded Java application connected to DayTrader benchmark.

4.1 DayTrader Benchmark

Modern CWS is usually built on middleware platforms such as J2EE. In this section, we describe used DayTrader performance benchmark, which is available as an open source application. Real CWS helps to identify configuration parameters. DayTrader is a suite of workloads that allows performance analysis of J2EE AS. It drives a trade scenario that allows to monitor their stock portfolio, inquire about stock quotes, buy or sell stock shares.

Tests have shown that the mean number of RPS for FE layer, on proposed hardware (one node in FE and one node in BE layer) and software (DayTrader) configuration, is about 1,400. Respectively, the mean measured number of RPS for BE layer CPU is about 4,000 RPS and for BE layer I/O about 7,500 RPS.

We can also see (Table 1) that the delay in the requests processing is mainly caused by the waiting time for service in the BE node in all cases (example for 30 clients), but the main problem is the performance of the system response time (System—one node in FE layer and one node in BE layer).

Table 1 Mean response time for: 20 RPS, 120 AS threads and 120 DBS connections

	Buy Quote[a]	Sell Quote	Update Profile	Show Quotes	Get Home	Get Portfolio	Show Profile
30 clients							
System (ms)	13.120	14.058	6.940	5.903	6.115	12.090	5.680
FE+BE (ms)	10.685	11.713	4.509	3.465	3.713	9.634	3.205
BE (ms)	10.202	11.224	4.229	3.083	3.056	8.733	2.959

[a] Buy Quote is only a few percent of all requests (Login/Logout → 4 %, Get Home → 20 %, Get Portfolio → 12 %, Show Quote → 40 %, Buy Quote → 5 %, Sell Quote → 5 %, Show Account → 10 %, Update Profile → 4 %)[1]

4.2 Cluster Experiments

Multiple FE servers (1, 2, and 3 nodes) and one BE node are the main configuration scenario. The structure—cluster environment—configuration is presented in table (Table 2). One of the most important requests (Buy Quote)[1] is used in tests 1, 2, and 3. All requests (All Quotes)—business transactions—shown in Sect. 4 are used in tests 4, 5, and 6.

Performance is measured in terms of the mean response time of business transactions. As we can see, corresponding tests have similar performance characteristics, but the mean response time for all requests—business transactions—(tests 4, 5, and 6) is less compared with Buy Quote requests—business transactions—(tests 1, 2, and 3).

Figure 1 shows the mean response time for more test cases (various number of clients' pools) also in multi-node environments with still increasing workload from 30 to 300 clients in 10 equal groups every 30 s. In the cases (with various number of clients' pools), we can observe that:

- a system consisting of one FE node (1 Node) in the tests (30–90 clients) has a response time similar to the system with two and three FE nodes (2 Nodes and 3 Nodes),
- a system consisting of three-node (3 Nodes) in the tests (120–180 clients) has a response time similar to the two-node system (2 Nodes),
- the difference between two-node and three-node systems—2 and 3 Nodes in the test over 270 clients is far smaller then in the tests for 210–270 clients.

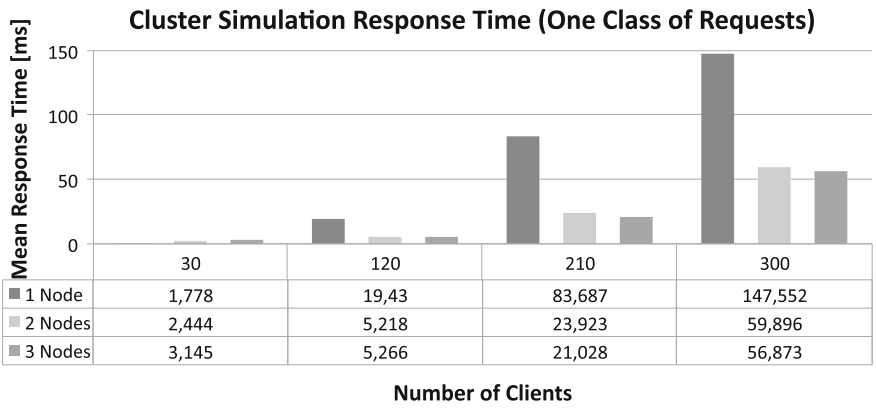
The results of the experiment on a three-node FE cluster show a smaller percentage increase in the overall system performance. Performance increased by 38 % with a second node added. By adding a next node, the system will be able to process only 13 % more requests.

Table 2 Input parameters of cluster experiments

Server/parameters	Test 1 (Buy Quote), Test 4 (All Quotes)	Test 2 (Buy Quote), Test 5 (All Quotes)	Test 3 (Buy Quote), Test 6 (All Quotes)
Client	10.10.10.1	10.10.10.1	10.10.10.1
GlassFish AS nodes	10.10.10.3	10.10.10.4–5	10.10.10.4–6
Oracle DBS node	10.10.10.2	10.10.10.2	10.10.10.2
AS threads pool	30	2×30	3×30
DBS connections pool	40	2×40	3×40
Number of RPS	15	15	15
Number of clients[a]	30, 120, 210, 300	30, 120, 210, 300	30, 120, 210, 300
Experiment time (s)	300	300	300

[a] Four subsets in all cases

[1] Requests class, which has the bigger impact on the behavior of the system (Table 1).



	30	120	210	300
1 Node	1,778	19,43	83,687	147,552
2 Nodes	2,444	5,218	23,923	59,896
3 Nodes	3,145	5,266	21,028	56,873

Fig. 1 Mean response time with different number of clients in clustered environments—still increasing workload

5 QPME Performance Analysis

The QPN model (Fig. 2) is used to predict the system performance. We use the QPME tool. Computational complexity was described in greater detail in [1].

Servers of the FE layer are modeled using the Processor Sharing (PS) queuing systems (*FE_CPU* places). The BE server is modeled by the following queues: PS modeling a processing unit (*BE_CPU* place) and First In First Out (FIFO) modeling a storage data (*BE_I/O* place) [2, 3]. *FE* and *BE* represent the places used to stop incoming requests when they await AS threads and DBS connections respectively. Clients are modeled by infinite server (IS) scheduling strategy (*Clients* place).

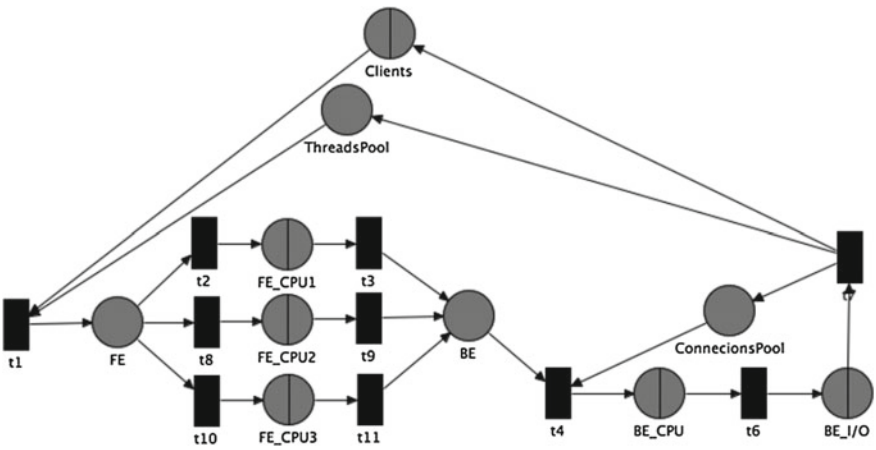


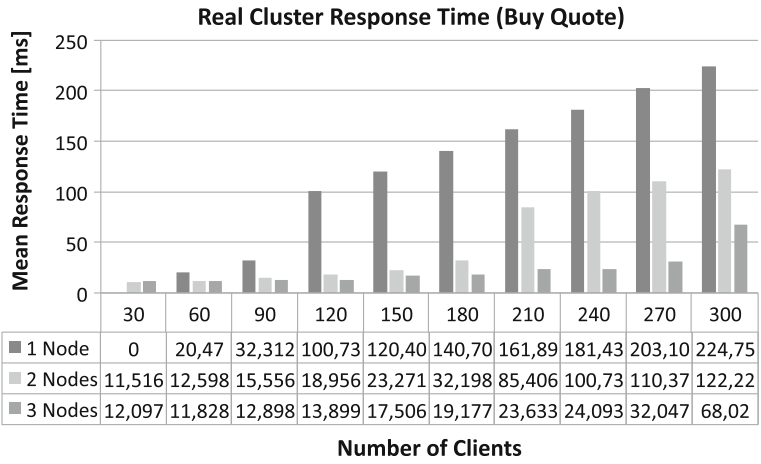
Fig. 2 System with FE cluster model

AS threads and DBS connections are modeled respectively by *ThreadsPool* and *ConnectionsPool* places (Fig. 2).

Software and client workload parameters are the same as in the experiment environment. Service demands (d) in layers based on experimental results in Sect. 4.2: $d_{FE\_CPU} = 0.714$ ms, $d_{BE\_CPU} = 0.25$ ms and $d_{BE\_I/O} = 0.133$ ms. Service in all queueing places is modeled by exponential distribution. Initial marking for places corresponds to the input parameters of cluster experiments (Table 2): Number of clients (number of tokens in *Clients* place), AS threads pool (number of tokens in *ThreadsPool* place), DBS connections pool (number of tokens in *ConnectionsPool* place). In these models, we have three types[2] of tokens: requests, AS threads, and connections to the DBS. The process of requests arrival to the CWS is modeled by exponential distribution with the λ parameter (client think time equal 66.67 ms) corresponding to the number of RPS (Table 2).

5.1 Cluster Simulation Results

The model developed in this section is validated by comparing its predictions against measurements in the real system. The results obtained in simulation modeling using QPME were compared with experiments based on DayTrader benchmark. Three application server nodes are available for the validation experiments. As expected (Fig. 3), overall response time decreases, while the number of nodes increases. However, in these simulations, response time (for various number of clients) differences between two-node and three-node systems are lower than between one-node system



	30	60	90	120	150	180	210	240	270	300
1 Node	0	20,47	32,312	100,73	120,40	140,70	161,89	181,43	203,10	224,75
2 Nodes	11,516	12,598	15,556	18,956	23,271	32,198	85,406	100,73	110,37	122,22
3 Nodes	12,097	11,828	12,898	13,899	17,506	19,177	23,633	24,093	32,047	68,02

Fig. 3 System with FE cluster simulation results

[2] A color specifying the types of tokens that can be reside in the place.

Table 3 Input parameters of simulations (one class of requests)

Model/parameters	Simulation 1	Simulation 2	Simulation 3
Load balancer	No	Yes	Yes
FE queues	*FE_CPU*1	*FE_CPU*1–2	*FE_CPU*1–3
BE queues	*BE_CPU*, *BE_IO*	*BE_CPU*, *BE_I/O*	*BE_CPU*, *BE_I/O*
ThreadPool place	30	60	90
ConnectionsPool place	40	80	120
λ	0.015	0.015	0.015
Number of clients	30; 120; 210; 300	30; 120; 210; 300	30; 120; 210; 300
Simulation time (s)	300	300	300

Table 4 Confidence interval for scenario with three-node in FE layer

Node	30 clients	120 clients	210 clients	300 clients
*FE_CPU*1	0.792	1.199	2.42	4.184
*FE_CPU*2	0.799	1.215	2.459	4.265
FE_CPU3	0.796	1.199	2.348	4.185
BE_CPU	0.278	0.437	0.963	4.118
BE_I/O	0.141	0.174	0.219	0.272

and two-node or three-node systems in the same way as in experiments. Some differences between the experiment and simulation results are due to the assumptions and the use of only one class of requests (Buy Quote) (Table 3).

The convergence of simulation results with the real systems results confirms the correctness of the modeling methods. The validation results show that the model is able to predict the performance for different number of nodes in the front-end layer. The modeling error does not exceed 20 %. A number of different models of realistic size and complexity were considered. The benchmark was run for 300 s per test, and each test was repeated 10 times to improve the reliability of results. The QPN model was simulated using the method of non-overlapping batch means method to estimate steady state mean token residence times. The average predicted response times are within the 95 % confidence interval of the measured average response times (Table 4). For all the simulations, the confidence intervals were sufficiently small for the results to be reliable. Our analysis showed that data reported by SimQPN is very stable.

6 Conclusions

We analyze (PE) the expected performance characteristics (response time) of a given system as a benchmark. We develop a PE framework for CWS that helps to identify

performance requirements. This class of systems does not allow simple parallelized processing, thus a solution to increase the systems efficiency is sought. The study demonstrates the modeling power and shows how the discussed models can be used to represent the system behavior.

In most cases, the results of real system tests are the same as those from performance simulation models. Despite the simplifications made, and on the assumptions given, it is possible to conduct an analysis corresponding with the modeled reality. The results demonstrate the effectiveness of the proposed modeling.

Future research will focus on verifying the system behavior and also the approximation of PE simulation and experiment results in case of a higher number of request classes used in simulations. The preliminary results show that adequate modifications can produce more acceptable level of compatibility between models and real systems. We also shall study the compromise between perceived average response time and energy consumption.

References

1. S. Kounev, *Performance Engineering of Distributed Component-Base Systems, Benchmarking, Modeling and Performance Prediction*, Shaker Verlag (2006)
2. T. Rak, S. Samolej, Simulation and performance analysis of distributed internet systems using TCPNs. Informatica **33**(4), 405–415 (2009)
3. T. Rak, S. Samolej, *Distributed Internet Systems Modeling Using TCPNs*, MCSIT (2008), pp. 559–566. doi:10.1109/IMCSIT.2008.4747298
4. S. Samolej, T. Szmuc, HTCPNs-based modelling and evaluation of dynamic computer cluster reconfiguration. LNCS **7054**, 97–108 (2012). doi:10.1007/978-3-642-28038-2_8
5. M. Szpyrka, P. Matyasik, R. Mrowka, Alvis-modelling language for concurrent systems. SCI, pp. 315–341 (2011). doi:10.1007/978-3-642-21271-0_15
6. K. Zatwarnicki, A. Zatwarnicka, Modeling operation of web service. CCIS **291**, 384–393 (2013). doi:10.1007/978-3-642-38865-1_39
7. T. Rak, J. Werewka, Performance analysis of interactive internet systems for a class of systems with dynamically changing offers. LNCS **7054**, 109–123 (2012). doi:10.1007/978-3-642-28038-2_9
8. E. Gelenbe, R. Lent, *Trade-offs Between Energy and Quality of Service* (IEEE Press, New York, 2012)
9. F. Bause, *Queueing Petri Nets-A formalism for the Combined Qualitative and Quantitative Analysis of Systems* (IEEE Press, New York, 1993), pp. 14–23. doi:10.1109/PNPM.1993.393439

Traffic Engineering: Erlang and Engset Models Revisited with Diffusion Approximation

Tadeusz Czachórski, Tomasz Nycz, Monika Nycz and Ferhan Pekergin

Abstract Classical Erlang and Engset formulae determining the availability of channels, loss probability, and characteristics of overflow traffic are still used in telecommunications. Moreover, they are also interesting for traffic management in mobile networks and in Internet. They are based on the assumption of Poisson flows and exponentially distributed time of connections. By means of diffusion approximation queuing models, we extend these results to the case of general distributions and transient state analysis.

1 Introduction

Queueing theory was started a hundred years ago by works of Agnar Krarup Erlang, an employee of Copenhagen Telephone Company, and by Tore Olaus Engset, traffic analyst and then director of Norvegian Televerket (now Telenor Group). Both of them were studying—in these days of human operators and cord boards to switch telephone calls by means of jack plugs—how many circuits were needed to provide an acceptable telephone service or how many telephone operators were needed to handle a given volume of calls. Their analysis [1–3] was based on Markov models, they

T. Czachórski (✉)
Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
Baltycka 5, 44-100 Gliwice, Poland
e-mail: tadek@iitis.pl

T. Nycz · M. Nycz
Institute of Informatics, Silesian University of Technology, ul. Akademicka 16,
44-100 Gliwice, Poland
e-mail: tomasz.nycz@polsl.pl

M. Nycz
e-mail: monika.nycz@polsl.pl

F. Pekergin
LIPN, Université Paris-Nord, 93 430 Villetaneuse, France
e-mail: pekergin@lipn.univ-paris13.fr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_26

assumed that the new connection demands made Poisson process and the duration of connections was given by a negative exponential distribution.

Erlang results were based on models that are called now, following Kendal's notation, queueing $M/M/c/c$ and $M/M/c$ models, see e.g. [4], where starting a connection is equivalent to an arrival of a customer and connection duration is customer's service time. The first model delivers an expression called Erlang B formula for blocking probability, i.e. probability $p(c)$ that all c parallel channels are occupied and the new calls are rejected. If the intensity of incoming Poisson stream is λ and the service intensity at each od the service channels is μ, then probability of n occupied channels is $p(n) = p(0)(1/n!)(\lambda/\mu)^n$, $n = 0, 1, \ldots, c$, probability $p(0)$ is given by normalisation, $\sum_{n=0}^{n=c} p(n) = 1$, and

$$p(c) = \frac{(1/c!)(\lambda/\mu)^c}{\sum_{n=0}^{n=c}(1/n!)(\lambda/\mu)^n} . \tag{1}$$

Erlang C formula gives probability that calls are queued, if their queueing is possible, i.e. $P_{wait} = P(n \geq c)$ in a $M/M/c$ system having c parallel channels and no limitations on the number of calls inside. In this system

$$p(n) = \begin{cases} p(0) \prod_{i=0}^{n-1} \frac{\lambda}{(i+1)\mu} = p(0) \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!} & \text{for} \quad n \leq c \\ p(0) \left(\frac{\lambda}{\mu}\right)^n \frac{1}{c!} \frac{1}{c^{n-c}} & \text{for} \quad n \geq c \, , \end{cases} \tag{2}$$

$p(0)$ is given by normalisation of probabilities, and

$$P_{\text{wait}} = \sum_{n=c}^{\infty} p(n) = \left(\frac{\lambda}{\mu}\right)^c \frac{1}{c!(1 - \frac{\lambda}{c\mu})}. \tag{3}$$

This solution is valid if $\frac{\lambda}{c\mu} < 1$, i.e. if the steady-state od this system exists.

Engset formula is equivalent to Erlang B formula in case where the population of potential connections is finite and limited to H connections: it is probability $p(c)$ at a $M/M/c/c/H$ system. Denote by ν the intensity of a single of call. If n calls are active, the total intensity of the remaining calls is $(H - n)\nu$. In this case, blocking probability $p(c)$ is

$$p(c) = \left[\binom{H}{c} (\nu/\mu)^c \right] / \left[\sum_{n=0}^{n=c} \binom{H}{n} (\nu/\mu)^n \right] \tag{4}$$

and the loss probability defined as the ratio of the lost calls stream to the offered calls stream

$$P_{\text{loss}} = \frac{(H-c)\nu p(c)}{\sum_{n=0}^{n=c}(H-n)\nu p(n)} = \frac{\binom{H}{c}(\nu/\mu)^c(H-c)\nu}{\sum_{n=0}^{n=c}\binom{H}{n}(\nu/\mu)^n(H-n)\nu}. \tag{5}$$

If all servers are occupied, the overflow traffic to be redirected to alternative paths is evaluated with the use of the distribution of customers outsizing c in a $M/M/c$ model. In particular, the mean and variance of this distribution is computed with the use of Riordan formulas [5]. There are also other approaches, see e.g. [6].

The significance of the above approach is not only historical, the cited formulae are currently in use in teletraffic engineering, also in mobile networks [7] and they are propositions to adapt them to modern Internet [8–10]. They result in more complex Markov models and application of numerical algorithms, e.g. Kaufman–Roberts recursion [11, 12] and convolution algorithms [13]. Therefore, to make the model more general, we present below the equivalent results obtained with the use of diffusion approximation. It allows us to consider non-Poisson input stream and non-exponential distributions of connections duration by applying $G/G/c/c$, $G/G/c/c/H$ models. Diffusion approximation allows us also to study transient states in case of time-dependent flows to see how the blocking probabilities vary with time and what is dynamics of the overflow traffic.

2 Diffusion Approximation Approach

The essence of diffusion approximation is the replacement of a stochastic process $N(t)$—the number of customers in a queueing system by a diffusion process $X(t)$. The diffusion equation (6) with appropriate parameters and boundary conditions determines the probability density function $f(x,t;x_0)$ of the process and this function is an approximation of the distribution of the number of customers in the service system.

$$\frac{\partial f(x,t;x_0)}{\partial t} = \frac{\alpha}{2}\frac{\partial^2 f(x,t;x_0)}{\partial x^2} - \beta\frac{\partial f(x,t;x_0)}{\partial x}. \tag{6}$$

We apply here the approach proposed by Erol Gelenbe in case of $G/G/1$ and $G/G/1/N$ stations [14] adapting it to multiple channel and finite population models where required diffusion parameters are state-dependent.

Let $A(x)$, $B(x)$ denote the interarrival and service time distributions. The distributions are general but not specified, the method requires only their two first moments: means $E[A] = 1/\lambda$, $E[B] = 1/\mu$ and variances $\text{Var}[A] = \sigma_A^2$, $\text{Var}[B] = \sigma_B^2$. Denote also squared coefficients of variation $C_A^2 = \sigma_A^2\lambda^2$, $C_B^2 = \sigma_B^2\mu^2$.

In a $G/G/1/N$ system, the choice of diffusion parameters is [14]: $\beta = \lambda - \mu$, $\alpha = \sigma_A^2\lambda^3 + \sigma_B^2\mu^3 = C_A^2\lambda + C_B^2\mu$. The processes $N(t)$ and $X(t)$ have then normally distributed changes with mean and variance increasing in the same way with the observation time.

In case of $G/G/1/N$ queue, the diffusion process should be limited to the interval $[0, N]$ corresponding to possible number of customers inside the system. To ensure it, two barriers are placed at $x = 0$ and $x = N$. In Gelenbe's model, when the diffusion process comes to $x = 0$, it remains there for a time exponentially distributed with parameter λ and then jumps instantaneously to $x = 1$. When the diffusion process comes to the barrier at $x = N$, it stays there for a time exponentially distributed with the parameter μ that corresponds to the time for which the queue is saturated and then jumps instantaneously to $x = N - 1$.

In case of multiple-service channels as in $G/G/c/c$ and $G/G/c$ stations, the output flow is state-dependent, and in case of finite population $G/G/c/c/H$ model also the input flow is state-dependent.

In $G/G/c/c$, we distinguish c subintervals: if $n - 1 < x < n$, it is supposed that n channels are busy (n customers inside the system), hence we choose

$$\alpha_n = \lambda C_A^2 + n\mu C_B^2, \qquad \beta_n = \lambda - n\mu \quad \text{for} \quad n - 1 < x < n, \qquad n = 1, 2, \ldots, c. \tag{7}$$

Jumps are performed from $x = 0$ to $x = 1$ with intensity λ and from $x = c$ to $x = c - 1$ with intensity $c\mu$.

Figure 1 shows the accuracy of this approach in case of steady-state analysis by comparing the Markovian model with its diffusion approximation and demonstrates how non-exponential distributions influence the results.

In case of $G/G/c$ system, the barrier at $x = c$ is removed and the last interval with parameters β_c, α_c is extended: $x \in (c - 1, \infty)$.

In case of $G/G/c/c/H$ system with finite population,

$$\beta_n = (H - n + 1)\nu - n\mu, \quad \alpha_n = (H - n + 1)\nu C_A^2 + n\mu C_B^2, \quad 1 \le n \le c,$$

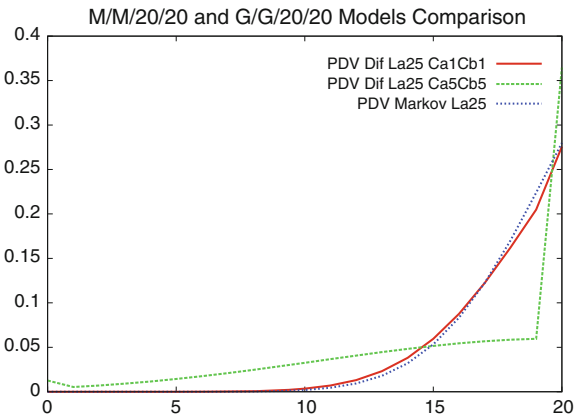


Fig. 1 A comparison of the distribution $p(n)$ at a M/M/20/20 model (Eq. 2) and its diffusion approximation, $\lambda = 25$, $\mu = 1$ and a corresponding G/G/20/20 model with $C_A^2 = C_B^2 = 5$

$$\beta_n = (H-n+1)\nu - c\mu, \quad \alpha_n = (H-n+1)\nu C_A^2 + c\mu C_B^2, \quad n \geq c.$$

where ν and C_A^2 refer to the sojourn time in the pool.

For the overflow traffic description, we may use probabilities $p(n)$, $n \geq c$ given by $G/G/c$ or $G/G/c//H$ models to extend Riordan formulae but it would be more natural to compute the characteristics of the flow in terms of diffusion models. If the pdf $f_A(x)$ of interarrival times mean $1/\lambda$ and variance σ_A^2 (squared coefficient of variation C_A^2 and $p(c)$ is the blocking probability, then the interevent density $f_{\text{overflow}}(x)$ of times between events in the overflow traffic is

$$f_{\text{overflow}}(x) = f_A(x)p(c) + f_A(x) * f_A(x)(1-p(c))p(c) + \cdots$$

giving mean $1/[\lambda p(c)]$ and squared coefficient of variation

$$C^2_{\text{overflow}} = p(c)\left(C_A^2 - 1\right) + 1.$$

These parameters may be used while splitting and merging traffic flows in the same way as it is done in network diffusion models [15].

Similarly, the pdf $f_{\text{acc}}(x)$ of interevent times at the accepted traffic is

$$f_{\text{acc}}(x) = f_A(x)(1-p(c)) + f_A(x) * f_A(x)p(c)(1-p(c)) + \cdots .$$

There is no probability transfer between intervals in steady-state solution, but we should take it into account in transient solution of $G/G/c/c$ and $G/G/c/c/H$ models. Inside each of c intervals of unitary length, the diffusion equation is solved assuming that the barriers at its left and right side act as absorbing ones. The density function $\phi(x, t; x_0)$ of a diffusion process limited by two absorbing barriers at $x = 0$ and $x = N$ ($N = 1$ in our case) and with the initial condition $x = x_0$ at $t = 0$ is, see e.g. [16]

$$\phi(x, t; x_0) = \frac{1}{\sqrt{2\Pi\alpha t}} \sum_{n=-\infty}^{\infty} (a_n - b_n)$$

where

$$a_n = \exp\left[\frac{\beta x_n'}{\alpha} - \frac{(x - x_0 - x_n' - \beta t)^2}{2\alpha t}\right], \quad b_n = \exp\left[\frac{\beta x_n''}{\alpha} - \frac{(x - x_0 - x_n'' - \beta t)^2}{2\alpha t}\right]$$

and $x_n' = 2nN, x_n'' = -2x_0 - x_n'$.

To balance probability flows between neighbouring intervals having different diffusion parameters, we put imaginary barriers between these intervals and suppose that the diffusion process which is entering a barrier at $x = n, n = 1, 2, \ldots, c-1$, from its left side (the process is increasing) is absorbed and immediately reappears at $x = n + \varepsilon$. Similarly, a process which is decreasing and enters the barrier from its

right side reappears at its other side at $x = n - \varepsilon$. The value of ε should be small, for example of the order of 2^{-10}, but we checked that it has no significant impact on the solution.

The density functions $f_i(x, t; \psi_i), i = 1, \ldots, c$, for the intervals $x \in]i-1, i[$ are as follows:

$$f_1(x,t;\psi_1) = \phi_1(x,t;\psi_1) + \int_0^t g_{1-\varepsilon}(\tau)\phi_1(x,t-\tau;1-\varepsilon)\mathrm{d}\tau\,,$$

$$f_n(x,t;\psi_n) = \phi_n(x,t;\psi_n) + \int_0^t g_{n-1+\varepsilon}(\tau)\phi_n(x,t-\tau;n-1+\varepsilon)\mathrm{d}\tau$$

$$+ \int_0^t g_{n-\varepsilon}(\tau)\phi_n(x,t-\tau;n-\varepsilon)\mathrm{d}\tau, \qquad n = 2,\ldots,c-1,$$

$$f_c(x,t;\psi_c) = \phi_c(x,t;\psi_c) + \int_0^t g_{c-1+\varepsilon}(\tau)\phi_c(x,t-\tau;c-1+\varepsilon)\mathrm{d}\tau \tag{8}$$

The relationships between the probability mass flows entering the barriers and reappearing at regeneration points are:

$$\gamma_n^R(t) = g_{n-\varepsilon}(t), \qquad \gamma_n^L(t) = g_{n+\varepsilon}(t), \qquad n = 1,\ldots,c-1 \tag{9}$$

with two exceptions concerning flows coming from barriers at $x = 0$ and $x = c$: $g_{1+\varepsilon}(t) = \gamma_1^L(t) + g_1(t)$, $g_{c-1-\varepsilon}(t) = \gamma_{c-1}^R(t) + g_{c-1}(t)$.

The system of equations is transformed with the use of Laplace transform and solved numerically to obtain the values of $\bar{f_n}(x, s; \psi_n)$. Then, we use the Stehfest inversion algorithm [17] to compute $f_n(x, t; \psi_n)$; for a specified t.

The parameters of the above model do not vary with time. However, we are interested in time-dependent input stream, hence the model is applied to small time-intervals, typically of one time-unit length, where the parameters are constant and the solution at the end of each interval gives the initial conditions for the next one.

3 Numerical Examples

Example 1 (extended Erlang model) Consider single $M/M/20/20$ and $G/G/20/20$ stations, at each channel $\mu = 1$. At $t = 0$, the system is empty. During the period $t \in [0, 20]$ the intensity of the input stream $\lambda_{in}(t) = 10$, then for $t \in [20, 50]$ $\lambda_{in}(t) = 25$, and for $t \in [50, 80]$ again $\lambda_{in}(t) = 10$.

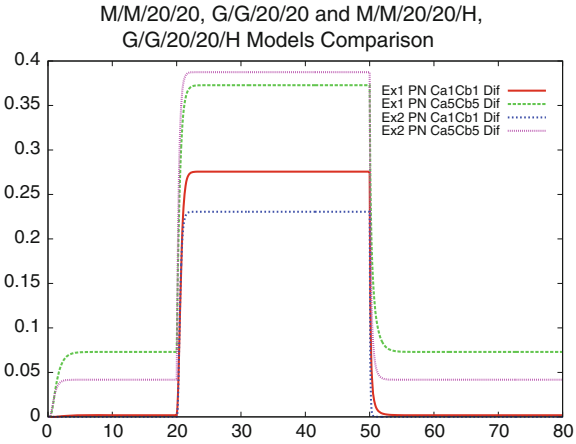


Fig. 2 Blocking probabilities as a function of time for both examples: finite and infinite population, $C_A^2 = C_B^2 = 1$ and $C_A^2 = C_B^2 = 5$,

Example 2 (extended Engset model) During the period $t \in [0, 20]$ there is a finite population of $H = 20$ connections, the activation intensity of each connection is $\nu = 1$. As previously, we consider a $M/M/20/20$ station, at each channel $\mu = 1$, and $t = 0$ the system is empty. Then in $t \in [20, 50]$ the population is $H = 40$, and for $t \in [50, 80]$ the size of the pool is again $H = 20$.

Figure 2 presents probabilities $p(c)$ for the both examples (Erlang and Engset models), for exponential and non-exponential distributions. The influence of large variations is more important in finite population model. During peak load ($t \in [20, 50]$ it is 70 % greater for $C_A^2 = C_B^2 = 5$ then for $C_A^2 = C_B^2 = 1$ in case of Engset model and 32 % greater in case of Erlang model. During low traffic periods, the relative difference of $p(c)$ in Markov and non-markovian models is much more distinct, as the value of $p(c)$ in Markov model is closed to zero.

We do not compare here the approximative results with simulation, but in general, the accuracy of these results is acceptable, see a discussion of diffusion approximation errors in [18].

4 Conclusions

The use of diffusion approximation allows us to extend Erlang and Engset results and to study the role of variances of call duration and times between calls distributions. It makes as well possible a transient analysis which is important because of constantly changing load in networks. A generalisation to multiple-service traffic (multiple flows with different intensities and of different duration) which is an is straightforward, as well as inclusion of threshold mechanisms differentiating demands in function of a current load.

Acknowledgments This work was supported by Polish project NCN nr 4796/B/T02/2011/40 *'Models for transmissions dynamics, congestion control and quality of service in Internet'* and the European Union from the European Social Fund (grant agreement number: UDA-POKL.04.01.01-00-106/09).

References

1. T.O. Engset, Die Wahrscheinlichkeitsrechnung zur Bestimmung der Whlerzahl in automatischen Fernsprechmtern. Elektrotechnische Zeitschrift, Heft 31, (1918)
2. A.K. Erlang, The theory of probabilities and telephone conversations. Nyt Tidsskrift Matematik, no. B **20**, 33–39 (1909)
3. A.K. Erlang, Solutions of some problems in the theory of probabilities of significance in automatic telephone exchnges. Electroteknikeren **13**, 5–13 (1917)
4. L. Kleinrock, *Queueing Systems*, vol. II (Wiley, New York, 1976)
5. R.I. Wilkinson, Teories of toll traffic engineering in the USA. Bell Syst. Tech. J. **40**, 421–514 (1956)
6. C. McArdle, D. Tafani, L. P. Barry, Overflow Traffic Moments in Channel Groups with Bernoulli-Poisson-Pascal (BPP) Load. in: *Proceedings of the IEEE International Conference on Communications* (IEEE ICC 2013), pp. 2403-2408, Budapest, Hungary, (2013)
7. M. Stasiak, M. Gabowski, A. Wisniewski, P. Zwierzykowski, Modelling and dimensioning of mobile networks, from GSM to LTE, Wiley (2011)
8. T. Bonald, J. Roberts, Internet and the Erlang formula. ACM SIGCOMM Comput. Commun. Rev. **42**(1), 25–30 (2012)
9. E. Chromy, J. Suran, M. Kovacik, M. Kavacky, Usage of Erlang formula in IP networks. Commun. Netw. **3**, 161–167 (2011)
10. J.W. Roberts, Traffic theory and the internet. IEEE Commun. Mag. **39**(1), 94–99 (2001)
11. J. Kaufman, Blocking in a shared resource environment. IEEE Trans. Commun. **COM–29**(10), 1474–1481 (1981)
12. J. W. Roberts, A service system with heterogenous user requirements—application to multiservice telecommunications systems, in: *Proceedings of the Performance of Data Communicatons Systems and their Applications*, ed. by G. Pujolle. Amsterdam, North Holland, pp. 423–431, (1981)
13. M. Glabowski, A. Kaliszan, M. Stasiak, Modeling product form state dependent systems with BPP traffic. J. Perform. Eval. **67**(3), 174–190 (2010)
14. E. Gelenbe, On approximate computer systems models. J. ACM **22**(2), 261–269 (1975)
15. E. Gelenbe, G. Pujolle, The behaviour of a single queue in a general queueing network. Acta Informatica **7**, 123–136 (1976). Fasc. 2
16. R.P. Cox, H.D. Miller, *The Theory of Stochastic Processes* (Chapman and Hall, London, 1965)
17. H. Stehfest, Algorithm 368: numeric inversion of laplace transform. Comm. ACM **13**(1), 47–49 (1970)
18. T. Czachorski, J.-M. Fourneau, T. Nycz, F. Pekergin, Diffusion approximation model of multiserver stations with losses. Electron. Notes Theor. Comput. Sci. **232**, 125143 (2009)

Part VI
Data Classification and Processing

CodeMagic: Semi-Automatic Assignment of ICD-10-AM Codes to Patient Records

Damla Arifoğlu, Onur Deniz, Kemal Aleçakır and Meltem Yöndem

1 Introduction

There is huge amount of data generated in modern hospitals, such as medical records, lab reports, and doctor notes. Health care systems employ a large number of categorization and classification systems to assist data management for a variety of tasks such as patient care, record storage and retrieval, statistical analysis, insurance, and billing. One of these systems is the International Classification of Diseases (ICD), which is the official system of assigning codes to diagnoses and procedures associated with hospital utilization. After a patient's clinical treatment, ICD codes are assigned to that patient's health records and insurance companies take these labels into account for reimbursement. On the other hand, label assignment by human annotators (clinic coders) may not be reliable. For example, in a Turkish hospital, two human experts examined 491 pre-labeled patient records and indicated that more than half of the records are assigned wrong ICD codes.

An ICD-10-AM code indicates a classification of a disease, symptom, procedure, or injury. Codes are organized hierarchically, where top-level entries are general groupings and bottom-level codes indicate specific symptoms or diseases and their location [1]. Each specific, low-level code consists of 4 or 5 digits, with a decimal

D. Arifoğlu (✉)
Sabanci University, Istanbul, Turkey
e-mail: damlaarifoglu@sabanciuniv.edu

O. Deniz · M. Yöndem
Middle East Technical University, Ankara, Turkey
e-mail: onurdeniz@gmail.com

M. Yöndem
e-mail: mturhan@ceng.metu.edu.tr

K. Aleçakır
Entegre Enformasyon Sistemleri, Ankara, Turkey
e-mail: kemalalecakir@ees.com.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_27

after the third. Overall, there are thousands of codes that cover a broad range of medical conditions [1]. Only trained experts can properly perform the task, making the process of coding documents both expensive and unreliable since a coder must select from thousands of codes for a given report. An accurate automated system would reduce costs, simplify the task for coders, and create a greater consensus and standardization of hospital data [1]. As a solution to this problem, we propose a recommender system that offers a set of ICD codes to given patient health records. Human annotators can thus select correct ICD codes in a shorter time since they have to choose ICD codes in a top k sets rather than thousands of codes. Please note that we work on the Tenth Revision, Clinical Modification (ICD-10-AM) of ICD and words *"Code"* or *"ICD code"* refer to this version throughout this study.

Medical datasets are usually characterized by their incompleteness and noisiness, which cause a substantial level of uncertainty while processing them [2, 3]. In Turkey, either doctors type reports themselves or medical assistants transfer dictated reports into text in hurry. This means no standardization through generation process of reports and many spelling errors which makes the processing of medical data challenging. The spelling error rate in Turkish medical records is 12–14 %, and tokenization is difficult in medical Turkish datasets [4]. In clinic documents, a variety of tokens is high, which makes these documents unstructured unlike radiology reports and processing them in a structured way becomes harder. Thus, our work also includes a medical-specific adjustment to common spell correction algorithms.

Search operations in medical information systems are currently carried out by classical information retrieval models. However, content in medical reports shows specific characteristics that a classic model cannot. For example, classical models decrease the significance of frequent words by weighting schemas or probabilistic methods, but they are the keywords bringing us that record's ICD code. For example, in a health record, which belongs to a patient treated in a cardiology department of a hospital, words such as *"liver"* are significant terms. We consider these types of domain specific words in our retrieval strategy by applying a boosting schema.

Since the Turkish is an agglutinative language, applying natural language processing techniques on Turkish documents is difficult. Any word may occur in corpus in its synonym or abbreviation forms. Variation in language use is not considered in classical retrieval models and results in lower recall values for retrieval tasks. We also consider these forms of original words in patient records during our retrieval strategy by constructing lexicons of synonyms and abbreviations.

Unlike most of the studies [5–7], our work not only focuses on radiology reports, but also many types of patient records such as surgery and clinical notes, which makes the automatic assignment of ICD codes even more difficult because of the availability of noise.

In this study, we focus on the problem of constructing an ICD-10-AM coding system for patient health records in Turkish hospitals. Since this labeling task requires expert knowledge in the field of medicine, process itself is costly and also prone to errors as the human annotators have to consider thousands of possible codes when assigning the correct ICD label to a document. This project arose as a need-based project since the Turkish Ministry of Health decided to use ICD labels in order to issue

payments to hospitals. We started our project after the meetings with ministry and clinical coders who are already working in Turkey's two biggest hospitals.[1] The aim here is to map a patient's already assigned diagnosis to an ICD code and represent all patient records with the identical standard and make the payment process less error prone.

To the best of our knowledge, our method is the first that suggests a solution to the problem of automatic assignment of ICD codes to patient records in Turkish and this study may be an inspiration to upcoming studies. We work on real data sets retrieved from hospitals in Turkey and our data sets are labeled by trained experts.

2 Related Work

Some examples of studies tackling the automatic extraction of ICD-9-CM codes can be found in the 2007 Challenge on Classifying Clinical Free Text [5]. Forty-four teams submitted their results to the challenge and several exploited the benefits of expert rules that were constructed either by experts in medicine, or by computer scientists. Only a small percentage of codes appear in the challenge. Increased documentation implies more false-negative ICD code matching, and thus, the greater difficulty of our task. Also radiology reports are written in exact templates rather than free text format so the grammatical structures used in these reports constitute a narrower subset of language and are easier to process grammatically. Moreover, our proposed method differs from these methods in terms of usage of training dataset since our method does not require any training dataset. Lastly, we work on ICD-10-AM, which contains about 68,000 codes and it becomes increasingly difficult to assign a correct ICD code to a given report.

Study [6] is one of the studies submitted to the challenge, in which the authors propose three different solutions. (1) Similarity of reports are calculated using Lucene search engine by exploiting individual words as queries. (2) After extracting n-grams (sequence of consecutive words) and s-grams (sequence of non-consecutive words) from reports, a boosting algorithm (BoosTexter) is applied to train classifiers without giving consideration to semantics. (3) A hand-crafted rule set is provided to capture lexical elements derived from BoosTexter's n-grams. Evaluations show that the third approach outperforms the other two methods since it explores the contribution of semantic features such as negations, synonyms, and uncertainty. Our proposed method shows some similarities with this method such as the usage of negation check, synonym/abbreviation extension, and using Lucene search engine.

In study [8], the task was considered as a multi-label text categorization problem. Maximum entropy modeling (MaxEnt) and support vector machine (SVM) were used to build the classifiers but treating the ICD code labeling problem as a classification problem does not make sense when the number of classes is so great. A similar

[1] We would like to thank to the Hospital of Ankara Numune Eğitim ve Araştırma and Hacettepe University Hospital.

approach is proposed in [9], where automatic matching of ICD codes to diagnoses in discharge letters is done by usage of support vector machines.

In study [1], a learning system labeling only 45 ICD codes by exploiting natural language features and a rule-based system that assigns codes based on the overlap between the reports is combined. Although studies show that rule-based systems are successful [6], solving the coding task using a set of hand-crafted expert rules has some drawbacks since the feasibility of the construction of such systems for thousands of ICD codes is indeed questionable.

In study [10], an ontology-based information extraction system for radiological reports is proposed. In this supervised approach, template-rules are defined in terms of ontology entities and syntactic regular expressions by a domain expert. The development of the first Turkish information extraction system is a contribution of this study.

3 Our Approach

We propose a recommendation system offering top k ICD codes for a given Electronic Health Record (EHR). Firstly, a given EHR is tokenized and normalized. Then, in a bag-of-words approach, each word's synonyms and abbreviations are added to the bags (named as *query bags*), while negation, temporal markers, and stop words are eliminated. We can consider these query bags as a token level vector space representation of the documents, thus as a feature set. Lastly, Lucene-based search strategy is used to extract all possible ICD-10-AM codes for each query bag and best candidate ICD codes are selected by Borda count voting schema.

3.1 Dataset

Given the fact that data collection and labeling is a time-consuming process (a clinical coder labels 20–25 reports in a day), we only focus on a subset of ICD-10-AM, specifically circulatory system. Moreover, we believe that a researcher need not to consider all clinical document sections to assign an ICD code to a health record; thus, we only extract the important patient-related events that occur within some fields.[2] We performed experiments on the following datasets, which were taken from Hacettepe University Hospital.

1. **All Department Dataset (ADD)**: A clinic coder chose 50 reports from 10 different hospital departments (such as cardiology, etc.) randomly and assigned ICD codes to them. This dataset has 500 reports in total.

[2] *"Patient History," "Surgery Notes," "Consultation," "Patient History," "Radiology," "Diagnosis"*.

2. **Partial Department Dataset (PDD)**: This dataset is a subset of the ADD dataset; constructed only with circulatory system clinical reports, thus it has 50 reports in total and the circulatory system refers to chapter I of ICD-10-AM.
3. **Gold Dataset (GD)**: As we mentioned before, since doctors fulfill clinical documents in a rush, these documents may be noisy and may not contain enough information to be assigned ICD codes. Also, given the nature of interpretation tasks and the subtle differences between ICD-10-AM codes, coders frequently disagree on the codes that should be assigned. We constructed a gold dataset so that two clinic coders chose 26 clinical documents and these documents contain enough information to be assigned at least one ICD code while having relatively less spell errors. Even with these documents, we see that some of the reports are given different ICD codes by different coders. Although these reports are said to be not ambiguous, out of 26, 16 reports are assigned different codes and 8 reports are assigned to even different chapters by coders.
4. **2-Level Dataset (2LD)**: ICD-10-AM codes may be consisted of 3–7 characters. Each level of subdivision after a category is a subcategory and the final level of subdivision is a code. Our observations show that assigning a correct ICD code grows increasingly difficult as the level of subdivision increases since the specificity of coding increases as well. While two clinic coders may assign different codes for a given EHR since the assignment of ICD codes is subjective, assigning an ICD code at all level characters is more challenging for an expert system. Thus, we aim the assignment of ICD codes at a second level in this study. Two clinic coders labeled 334 clinic documents from a circulatory chapter at a second level (e.g., They labeled I25 instead of I25.5),

3.2 Preprocessing

Before the code assignment task, practical problems such as segmentation, tokenization, and spelling errors of unstructured medical reports have also been addressed. Since reports are generated via unstructured environments, markers of report parts vary with the user who generate the report. Case differences, number of space characters after/before tokens, and different markers for same parts are observed along the dataset. We used regular expressions in order to cope with these differences. After tokenization and labeling, normalization of tokens—clearing successive and redundant punctuation and spaces, separating words which are accidentally combined with punctuation (e.g., “aittir.Sol”) or uniting accidentally separated words with punctuation—is performed using regular expressions. Then, reports are segmented into sentences and then into words using regular expressions. Given the fact that using the roots of terms increases efficiency of information retrieval tasks, we stem each tokenized word. For this purpose, we used Zemberek [11] and added terminological roots to its present lexicon. Spell correction using Levenshtein distance is used for typing-based errors (see [4] for detail).

3.3 Query Formation

Articles, prepositions, conjunctions, adverbs, and any other common words, which are too common among documents, are referred as stop words and such words are removed before the search process. Moreover, some keywords are more important to reach the correct ICD code, so we boost these words during our retrieval strategy. For this purpose, clinical coders constructed lexicons for stop words and important words for circulatory (I) chapter of ICD-10 CM.

Although ICD-10-AM guides contain many useful synonyms, the coverage of these guides is not perfect. There are expressions and abbreviations, which are characteristic of a particular health institute where the document was created. Given the idea that the better systems process negations, hypernyms, and synonyms [9], to construct a synonym and abbreviation list, we examined about 1,000 health reports. We firstly tokenize the words in reports and these words are stemmed by using Zemberek [11]. Then, two clinic coders manually assigned synonyms and abbreviations for those words to construct a lexicon.

Because of the procedural examinations, sentences in clinical reports have both negative and positive polarity values. In other words, even if the finding of a body structure is natural, the normality of corresponding feature is still expressed. However, these kind of sentences with negative polarity in which normality is expressed are still indexed in classic IR models with default document representation of list of terms. In study [12], it is noted that the information contained in the negative phrases of a diagnosis is lost when using a bag-of-words approach to mine a medical text, which then results in negated items being erroneously considered as if present in the patient. Inspired by this, in order to deal with such kind of scenarios, our document representation is updated so that positive and negative polarity values are handled differently. In Turkish, a negative marker on verbal phrases, *-ME/-MA*, or the verb *"yoktur" or "değildir"* gives negative polarity to a sentence. Thus, if a sentence contains one or more of these phrases, that sentence is removed from query list since it has negative polarity. Moreover, in a clinical report, a treatment that was applied to the patient many years ago may be mentioned, although it has nothing to do with the current diagnosis. So we constructed a temporal marker list to remove these sentences from our query bags.

By exploiting the idea of bag-of-words, each tokenized word is added to a bag and this bag is named as a *query bag*. A drawback of this algorithm is that word order of this sentence and contextual information ignored. Given the idea that a word in a clinical report may occur in its synonym or abbreviation version, we added that word's synonym and abbreviations in our lexicon while removing stop words. Also if the query bag has a temporal or negation marker, we remove that query bag from our query list and we boost important words. We use Lucene's OR search strategy during retrieval process, so we connect a word to its synonym or abbreviations by using OR tokens.

3.4 ICD Code Retrieval

The ICD-10-AM is divided into the alphabetic index (AI), an alphabetical list of terms and their corresponding codes. ICD expanded index is constructed by traversing the AI tree and finding words describing each ICD code. Before the retrieval of words in query bags, ICD expanded index entries are indexed with Lucene search engine. Then, each query bag is matched with ICD expanded index entries and thus brings a list of codes, which are ranked by their Lucene match scores. Then, these codes will be re-ranked by Borda count voting method.

We used Lucene tool [13] for indexing ICD-10-AM expanded index entries and searching for query bags. Given query bags, Lucene search engine retrieves candidate ICD codes by matching words in query bags and ICD expanded index entries. Each query bag brings some candidate ICD codes in an ordered list, which means each query bag has a preference list of candidate ICD codes. Then, Borda count method [14] is used to re-rank these preference lists. In the end, a top k-NN list is constructed as a recommendation list of codes.

Assume that we have n query bags in a given health record. Each query bag is matched with entries in an ICD expanded list, so each query bag brings a list of candidate ICD codes ranked by their Lucene scores, because each entry is mapped to an ICD index. These ICD codes are ranked by their matching score, which is calculated by Lucene search engine based on the occurrences of query bag words in the entries. Let us say a query bag i retrieved ICD codes in the order of $L = \{c_{i1}, c_{i2}, ..., c_{im}\}$, once this list is ranked by their Lucene score, we are not interested in their exact scores since we will assign Borda count scores for them. Note that for each query bag, we only take their top k ranking candidate ICD codes (we set k to 500 in our experiments). Thus for list L, Borda count scores will be $S = \{k-1, k-2, ..., k-k\}$ correspondingly. At the end, for each query bag, we have codes and their corresponding Borda count scores. Then, for each unique ICD code, we sum its Borda count scores coming from different query bags. Then, all unique ICD codes are re-ranked by their new scores and a ranking list of ICD codes is offered to the user.

4 Experiments

Considering ICD code matching as a clustering problem, rather than searching the possible clustering by dividing the whole set into some cluster, a constructive bottom up approach is used. We observed that it would be easier to firstly map a clinical document to its ICD chapter and then level by level, to a correct ICD code. Given all 26 ICD chapters, we firstly select strong chapter candidates for a given EHR by filtering out the weaker ones. For this purpose, we firstly retrieve ICD codes for each query bag in a document. Then for each code, its chapter (e.g., code I25.10 belongs to chapter I) is counted for a voting in the Borda count method and a top chapter list is provided by considering first N candidate ICD chapters (we set N to 500).

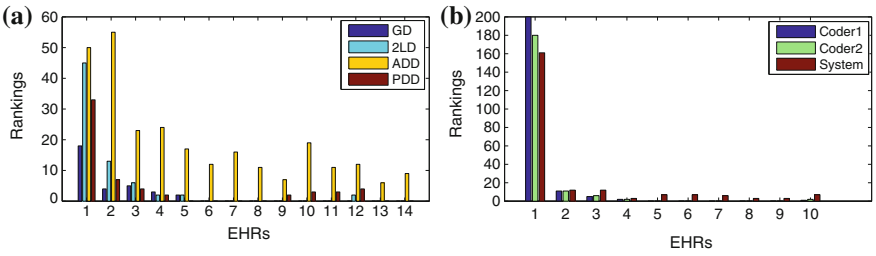


Fig. 1 **a** Chapter mapping ranking histograms for each data set. **b** 2LD chapter mapping ranking histograms when labels by two different coders and the system are used as ground truth

In Fig. 1a, we see the distribution of chapter mapping rankings. While our method is successful to assign correct chapters in the best 5 chapter candidates for datasets GD and 2LD, it is not the case for ADD since this dataset contains sample reports from other chapters as well and remember that we prepared boosting, stopping words, abbreviation, and synonyms word lexicons only for chapter I. This analysis also shows the effectiveness of these word lists and when the system is boosted with these lexicons, success increases.

Remember that the 2LD dataset was labeled by 2 different coders and we already have its labels coded by experts in the hospital. Since detailed reading is necessary to reach a correct code, codes that were assigned by the hospital are not reliable. In Fig. 1b, chapter mapping ranking histograms are given for the 2LD dataset when 3 different label sources are used: coder 1, coder 2, and system (already coded by experts in the hospital, but in rush). Our domain expert observed that clinical coder 1 is more trustworthy in correct labeling and we see that our methods success is higher when the first coder's labels are taken as a baseline.

There may be 3 different types of diagnoses for a given patient data: primary diagnosis, secondary diagnosis, and a possible diagnosis. In a hospital, a clinical coder assigns ICD codes for all these diagnosis types. Although only one ICD code can be assigned as a primary diagnosis, many secondary diagnosis codes may be given to a EHR. A possible diagnosis describes all other diagnoses that a patient may have received treatment. In Fig. 2a, chapter mapping ranking analysis is done when all diagnosis types are considered. Remember that this dataset is constructed by clinical documents from circulatory system. Thus, primary diagnoses belong to the chapter I of ICD-10-AM and our method offers this chapter codes at the first ranking. On the other hand, secondary diagnoses may be coming from different chapters, so our I-chapter-adapted system may not offer their chapter ids at the first rank.

After chapter mapping is done, we only consider candidate codes belonging to the first 5 candidate chapters offered in the previous step for each EHR. Instead of exact matching, we decided to do partial matching at the second level. For example, for a clinical document which has a correct ICD code I50.33, we aim to offer candidate codes at a second level (e.g., I50). While it is nearly impossible for a human expert to assign an ICD code at an exact level to a given EHR, developing tools to automatize this process are not realistic.

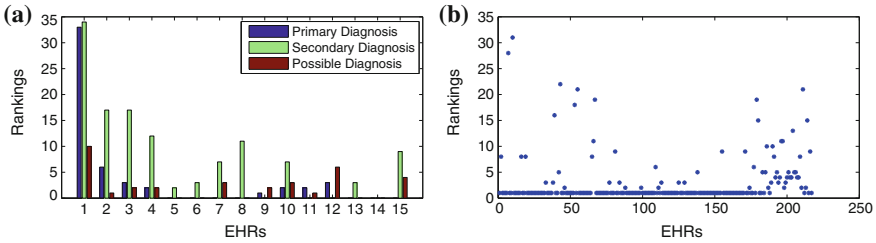


Fig. 2 **a** PDD chapter mapping histograms when all diagnosis types are considered during the test. The X-axis represents the ranking of correct chapters of clinical documents, while the y-axis is the number of EHRs that fall into that ranking bin. **b** Partial data set exact matching rankings for primary diagnoses. We see that most of the codes are offered at a very top ranking

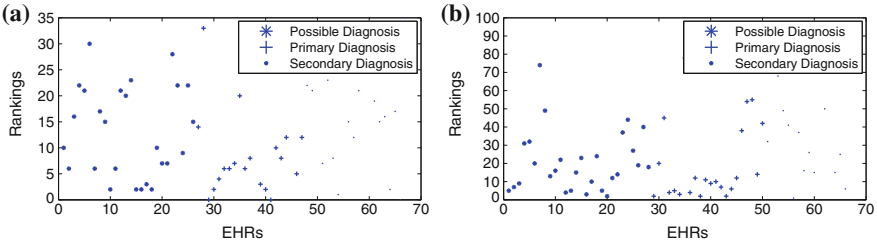


Fig. 3 **a** Second level exact matching rankings on gold dataset. **b** All level exact matching rankings on gold dataset. Primary diagnosis are offered at lower ranks while secondary diagnosis are more spread out, since all primary diagnosis belong to chap. 1 while secondary diagnosis and possible diagnosis may come from any chapters and our system is onsly adapted for chap. 1

As it can be seen in Fig. 3a, b primary diagnoses can be retrieved at very low rankings, but it is not the case for secondary diagnoses. This happens even with for a human expert since a given report generally contain more information about a primary diagnosis than does a secondary. In Fig. 3b, we see exact matching hits for our gold dataset and again for primary codes are retrieved at top rankings.

5 Conclusion and Future Work

In this preliminary study, we proposed a recommender system to offer top k ICD codes for a given patient record. Our solution exploits the idea of bag-of-words approach using a Lucene search engine and Borda count method. We also propose solutions for preprocessing such as tokenization and spell checking. Our experiments on a set of patient records show that our method gives promising results for automatically tagging clinical findings with ICD-10-AM codes. In future, we are planning to index semantic concepts in the document so that we can capture the semantic relationships between words.

Acknowledgments This project is funded by EES (a software company in Turkey) and TÜBİTAK (Research Council of Turkey) under grant number 3110502.

References

1. K. Crammer, M. Dredze, K. Ganchev, P.P. Talukdar, S. Carroll, Automatic code assignment to medical text, in *Proceedings of the Workshop on BioNLP 2007: Biological, Translational and Clinical Language Processing*, pp. 129–136 (2007)
2. N. Lavrac, Selected techniques for data mining in medicine, in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 16, p. 23 (1999)
3. K.J.M. Janssen et al., Missing covariate data in medical research: to impute is better than to ignore. J. Clin. Epidemiol **63**, 721–727 (2010)
4. O. Deniz, Ontology based text mining in Turkish radiology reports (Master's Thesis, Middle East Technical University, Turkey, 2011)
5. 2007 International Challenge: Classifying clinical free text using natural language processing
6. I. Goldstein, A. Arzumtsyan, O. Uzuner, Three approaches to automatic assignment of ICD-9-CM codes to radiology reports, in *Proceedings of the Fall Symposium of the American Medical Informatics Association*, Chicago, Illinois, USA, American Medical Informatics Association, pp. 279–283 (2007)
7. R. Farkas, G. Szarvas, Automatic construction of rule-based ICD-9-CM coding systems. BMC Bioinformatics **S–3**, 9 (2008)
8. J. Patrick, Y. Zhang, Y. Wang, Developing feature types for classifying clinical notes, in *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pp. 191–192 (2007)
9. S. Boytcheva, Automatic matching of ICD-10 codes to diagnoses in discharge letters, in *Proceedings of the Second Workshop on Biomedical Natural Language Processing*, pp. 19–26 (2011)
10. E. Soysal, I. Cicekli, N. Baykal, Design and evaluation of an ontology based information extraction system for radiological reports. Comput. Biol. Med. **40**, 900–911 (2010)
11. A.A. Akin, M.D. Akin, Zemberek, an open source NLP framework for Turkic languages. (2007)
12. R.B. Rao, S. Sandilya, R.S. Niculescu, Clinical and financial outcomes analysis with existing hospital patient records, in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 416–425 (2003)
13. D. Cutting, Lucene search engine. (1999) http://lucene.apache.org/
14. M.V. Erp, L. Schomaker, Variants of the borda count method for combining ranked classifier hypotheses, in *The Seventh Intenational Workshop on Frontiers in Handwriting Recognition*, pp. 443–452 (2000)

Utilizing Coverage Lists as a Pruning Mechanism for Concept Discovery

Alev Mutlu, Abdullah Dogan and Pinar Karagoz

Abstract Inductive logic programming (ILP)-based concept discovery systems lack computational efficiency due to the evaluation of the large search spaces they build. One way to tackle this issue is employing pruning mechanisms. In this work, we propose a two-phase pruning mechanism for concept discovery systems that employ an Apriori-like refinement operator and evaluate the goodness of the concept descriptors based on their support value. The first step, which is novel in this work, is computationally inexpensive and prunes the search space based on the coverages of the concept descriptors. The second step employs a widely employed pruning mechanism based on the support value of the concept descriptors. The experimental results show that the first step leaves a search space reduced by 4–22 % to be evaluated by the second step, which is more costly.

Keywords Concept discovery · Pruning · Support · Coverage lists

1 Introduction

Concept discovery is one of the most commonly addressed tasks in multi-relational data mining [1]. It aims to induce complete and consistent definitions of a relation, called the *target relation*, in terms of other relations, called the *background knowledge*. Most of the concept discovery systems have their roots in ILP [2], as it provides an expressive representation framework, namely first-order logic, for data stored in multiple tables. Although such a representation framework enables concept discov-

A. Mutlu (✉)
Department of Computer Engineering, Selcuk University, Konya, Turkey
e-mail: alevmutlu@selcuk.edu.tr

A. Dogan · P. Karagoz
Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
e-mail: adogan@ceng.metu.edu.tr

P. Karagoz
e-mail: karagoz@ceng.metu.edu.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_28

ery systems to induce high-quality and human interpretable concept descriptors, it also builds large search spaces that concept discovery systems need to evaluate. Evaluation of such large search spaces introduces computational efficiency and limits the applicability of ILP-based concept discovery systems.

Several methods such as memoization [3], parallelization [4], query optimization [5], and sampling [6] have been studied to speedup the search space evaluation process and promising results are reported. Another direction to cope with the large search space is the employment of pruning mechanisms. Pruning is a way of reducing the size of a search space by removing some portion of it. It is an efficient way to reduce the size of the search space, though caution should be paid not to remove sections that contain valuable information. In concept discovery systems, pruning mechanisms based on confidence and support [7], language bias [8], and generality ordering [9] have been studied.

In this work, we propose a two-phase pruning mechanism for ILP-based concept discovery systems that employ an Apriori-like refinement operator and prune the search space based on the support value of the concept descriptors. This first step of the proposed pruning, which is novel in this work, is computationally inexpensive and is based on the coverage set of the concept descriptors. It assumes that if two hypotheses are refined via an Apriori-like operator, the coverage set of a refined hypothesis should at maximum be the intersection of the coverage sets of its parents. The second step of the pruning, which is computationally expensive, is based the support values of the concept descriptors.

The proposed pruning mechanism is implemented as an extension to a concept discovery system called Tabular CRIS wEF [10] and is evaluated on different data sets. The experimental results show that the first step of the proposed pruning mechanism reduces the search space by 4–22 %, depending on the data set, and completely preserves the accuracy of the induced concept descriptors.

The rest of this paper is organized as follows: Sect. 2 we introduce Tabular CRIS wEF as the proposed algorithm is embodied into it. In Sect. 3, we present the pruning algorithm. In Sect. 4, we present the experimental results, and in Sect. 5 the paper is concluded an overview and final remarks.

2 Tabular CRIS wEF

Tabular CRIS wEF is a predictive ILP-based concept discovery system. The concept descriptor induction process in Tabular CRIS wEF starts with the formation of the most general two-literal concept descriptors, and iteratively these concept descriptors are refined via relational version of the Apriori refinement operator (1) to form the final concept descriptors. To incorporate the background examples that are indirectly related to the target instances, Tabular CRIS wEF performs a further specialization process on the existential variables of the same type in the body of the clause.

$$C_1 \cup C_2 = \{C_1 \cup l_{21} | C_{12} = C \cap C_2 - C_{21} = l_{21}\} \tag{1}$$

At the end of each epoch Tabular CRIS wEF employs the following pruning mechanism:

1. If both support (2) and confidence (3) values of a concept descriptor are higher than the minimum values it is removed from the search space and added to the solution set.
2. If a concept descriptor has a support value less than the threshold but confidence value higher than both of its parents it is further refined.
3. If a concept descriptor has a confidence value less then either of its parents it is pruned.

$$\sup(h \leftarrow b) = \frac{|\text{bindings of variables for h that satisfy h} \leftarrow \text{b}|}{|\text{bindings of variables for h that satisfy h}|} \tag{2}$$

$$\text{conf}(h \leftarrow b) = \frac{|\text{bindings of variables for h that satisfy h} \leftarrow \text{b}|}{|\text{bindings of variables for h that satisfy b}|} \tag{3}$$

Once the concept descriptors reach the maximum length, target instances that are modeled by the concept descriptors in the solution set are removed from the target instance set and induction process restarts with the remaining target instances.

Due to the refinement operator, Tabular CRIS wEF might produce different concept descriptors that map into the same evaluation query. In order to prevent the execution of the same query multiple times, Tabular CRIS wEF stores each evaluation query and its result in a hash tables, that is it has a memoization capability. Whenever a previously executed evaluation query is generated, its result set is retrieved from the hash table. The stored results of the evaluation queries are updated as the covering method is executed. The concept descriptor induction process stops once the number of target instances goes below a certain threshold or no more concept descriptors can be generated to explain the remaining target instances.

3 The Proposed Pruning Algorithm

The motivation behind the proposed pruning algorithm is that if two concept descriptors are refined through an Apriori-like refinement operator and are pruned based on their support values, the coverage set of the refined concept descriptor that counts for its support may at maximum be the intersection of its parents' coverage sets.

Noting that support of a concept descriptor to is the cardinality of the set of target instances it models, this motivation can be explained in analogy to the join operator of relational algebra. The join of relations r and s, such that $r(R)$, $s(S)$ with $RS = T$, is the relation $q(T)$ of all tuples t in over T such that there are tuples $t_r \in r$ and $t_s \in s$ with $t_r = t(R)$ and $t_s = t(S)$. Since $R \cap S$ is a subset of both R and S, as a consequence of the definition $t_r(R \cap S) = t_s(R \cap S)$. Thus, every tuple in q is a combination of a tuple from r and a tuple of s with equal $(R \cap S)$-values [11].

In the case of Apriori-based refinement of two concept descriptors, say C_1 and C_2, coverage of C_1 maps to $r(R)$ and coverage of C_2 maps to $s(S)$, support coverage of the refined concept descriptor is a subset of $(R \cap S)$-values.

Although the idea seems straightforward to implement there are certain cases to be considered. The refinement operator generalizes each argument of the predicates in all possible ways to incorporate the background data and not to miss any concept descriptor. The following all three concept descriptors of length 4

C_4^1: elti(A, B):-brother(C, D), husband(C, A), husband(E, B)

C_4^2: elti(A, B):-brother(C, D), husband(C, A), husband(C, B)

C_4^3: elti(A, B):-brother(C, D), husband(C, A), husband(D, B)

are refinements of following two concept descriptors of length 3.

C_3^1: elti(A, B):-brother(C, D), husband(C, A)

C_3^2: elti(A, B):-husband(C, A), husband(D, B)

In these refinements, the first two literals come from C_3^1 and the last literal comes from C_3^2. With a closer inspection on the argument properties of the literals, i.e., being bound to head or to a some other body literal argument, shows that support coverage set of C_4^1 may directly be obtained by intersecting the support coverage set of the parent clauses as the first argument of the third body literal, namely E, is unbound both in C_3^2 and C_4^1. On the other hand, situation is different with C_4^2 and C_4^3.

In C_4^2 the first argument of the last body literal, which is an unbound argument in the parent clause, is bound both to the first arguments of the first and the second body literals. Similarly, again the the same argument in C_4^3 is bound to the second argument of the first argument of the first body literal. In such cases simply intersecting support coverage set of parent clauses will not produce correct sets.

In order to handle such cases, the proposed algorithm extracts parent concept descriptors from the refined concept descriptors with the same binding properties. For C_4^2 these parents are

C_3^3: elti(A, B):-brother(C, D), husband(C, A)

C_3^4: elti(A, B):-brother(C, D), husband(C, B)

and for C_4^3 are

C_3^5: elti(A, B):-brother(C, D), husband(C, A)

C_3^6: elti(A, B):-brother(C, D), husband(D, B)

To find the correct support coverage set intersection, the proposed pruning algorithm intersects C_3^1 with C_3^4 and C_3^1 with C_3^6, respectively, for C_4^2 and C_4^3. The pruning algorithm does not need to evaluate support coverage sets of C_3^4 and C_3^6 as they are already evaluated in the previous iterations and their support coverage sets are stored in the memo tables.

Algorithm 1 outlines the refinement and the pruning mechanism of Tabular CRIS wEF. It inputs an array concept descriptors sorted in lexicographical order. If two

clauses are unifiable they are unified and the refined clauses are pruned based on coverage sets of their parents. Extracting the parent concept descriptors from a refined concept is an easy task as each concept descriptor stores which literally comes from which parent clause and its position in that concept descriptor.

If they pass this first pruning step, then they go through the second pruning where they are pruned based on their support and confidence values.

Algorithm 1 refineAndPrune(vector<possibleClauses> pC)

```
1: for (i = 0; i < pC.size() - 1; i++) do
2:   for (j = i+1; j < pC.size(); j++) do
3:     if (unifiable(pC[i], pC[j])) then
4:       tmp_pC = unify(pC[i], pC[j]);
5:       for (k = 0; tmp_pC.size(); k++) do
6:         parent[k] = getParent(k);
7:       end for
8:       support_set = set_intersect(parent);
9:       if (support_set.size() ≥ min_set_size) then
10:        pruneFurther(tmp_pC[k]);
11:      end if
12:    end if
13:  end for
14: end for
```

We would also like to note that the proposed pruning mechanism does not require to store any information other than Tabular CRIS wEF already stores.

4 Experiments

In order to evaluate the performance of the proposed pruning mechanism, we conducted a set of experiments on different data sets. In Table 1 we list the properties of the data sets and the experimental setting. *Dunur*[1,2] and *Elti*[3,4] [7] are kinship data sets, *Muta* [12] and *PTE* [13] are biochemical data sets. *Muta Small* is a subset of the Muta data set and extended with aggregate predicates. Similarly, *PTE 5 Aggr.* is the extended version of PTE data set with aggregate predicates. These data sets contain information about the mutagenicity and carcinogenicity of drugs and chemicals. *Student Loan* [14] contains data about students enrollment and employment status,

[1] Dunur is a relatedness of two people due to marriage such that A is dunur of B if a child of A is married to a child of B.

[2] It is a symmetric relation.

[3] Elti is a relatedness of two people due to marriage such that A is elti of B if A's husband is brother of B's husband.

[4] See footnote 2.

Table 1 Data set properties and experimental settings

Data set	Num. relations	Num. facts	Min. sup.	Min. conf.	Length
Dunur	9	234	0.3	0.7	4
Elti	9	234	0.3	0.7	4
Muta small	8	274	0.3	0.7	4
Muta	8	16,544	0.3	0.7	4
PTE No Aggr.	27	29,267	0.1	0.7	4
PTE 5 Aggr.	32	29,267	0.1	0.7	4
Student loan	10	5,288	0.1	0.7	4

and the aim is to find rules that define a students' obligation for paying his/her loan back.

The code is developed in C and the experiments are conducted on a PC with Intel® Core ™ i7-2600K CPU processor and 7.8 GB ram.

In Table 2 we report the experimental results. The second and the fourth columns compare the number of concept descriptors generated, and the columns three and five compare the running time of the systems with the pruning time embodied and not, respectively. As the experimental results show the proposed pruning mechanism reduces the size of the search space by 10 % on the average.

$$\text{sensitivity} = \frac{\text{true\_positive}}{\text{true\_positive} + \text{false\_negative}} \tag{4}$$

$$\text{specificity} = \frac{\text{true\_negative}}{\text{false\_positive} + \text{true\_negative}} \tag{5}$$

$$\text{PPV} = \frac{\text{true\_positive}}{\text{true\_positive} + \text{false\_positive}} \tag{6}$$

$$\text{NPV} = \frac{\text{true\_negative}}{\text{true\_negative} + \text{false\_negative}} \tag{7}$$

In order to evaluate the success of the proposed pruning method, we report the sensitivity, specificity, positive predictive value (PPV), and negative predictive value (NPV) of pruning method for the experiments. Sensitivity (4) is a goodness measure of a predictor, in our work the pruning mechanism, to identify a condition correctly; and the specificity (5), on the other hand, is a goodness measure of a predictor to exclude a condition. PPV (6) and NPV (7) describe the performance of a diagnostic test. PPV refers to the proportion of people with a positive test result who actually have the disease, and NPV refers to the proportion of people with a negative test result who do not have disease [15]. The higher values of PPV an NPV suggest that the predictor is doing as good as the gold standard [16].

Table 2 Experimental results

Data set	Without pruning		With pruning		Change (%)
	# Concept descriptors	Running time (ms)	# Concept descriptors	Running time (ms)	
Dunur	1,887	2,348	1,462	2,055	22.5
Elti	1,741	2,328	1,540	2,202	11.5
Muta small	6,056	12,682	5,781	12,519	4.5
Muta	62,486	2,080,472	56,752	2,035,056	9.2
PTE	11,166	142,951	10,121	116,794	9.4
PTE 5 Aggr.	64,322	2,017,402	55,729	1,950,077	13.4
Student loan	305,282	777,713	286,270	762,048	6.2

Table 3 Predictive evaluation of the pruning method

Data set	Sensitivity	Specificity	PPV	NPV
Dunur	0.53	1	1	0.70
Elti	0.27	1	1	0.60
Muta small	0.13	1	1	0.66
Muta	0.28	1	1	0.72
PTE No Aggr.	0.38	1	1	0.61
PTE 5 Aggr.	0.42	1	1	0.78
Student loan	1	1	1	1

We report these results in Table 3. As the results show the sensitivity of the proposed pruning mechanism is 0.43 on the average and specificity is 1, which means that the proposed method never identifies a concept descriptor with high coverage value as a concept descriptor with a low coverage value, but may predict concept descriptors with low coverage value as ones with high coverage value. Indeed, these results comply with the motivation of the study: eliminating as many concept descriptors with potential low coverage value as possible and none of the concept descriptors with potential high coverage value. Also the PPV value is 1, which is the gold standard, and NPV value is 0.67 on the average.

5 Conclusion

In this work, we propose a two-phase pruning mechanism for ILP-based concept discovery systems that employ Apriori-like refinement operator and prune the search space according to thecoverage lists coverage of the concept descriptors. The motivation behind the pruning mechanism is that coverage set of a concept descriptor may at maximum be the intersection of the coverage sets of its parents. The proposed

pruning mechanism is computationally inexpensive and reduces the search space by 4–22 % for the evaluated data sets without causing any loss in the accuracy of the solution clauses. The statistical analysis of the results shows that they are statistically significant.

References

1. S. Dzeroski, Multi-relational data mining: an introduction. SIGKDD Explor. **5**(1), 1–16 (2003)
2. S. Muggleton, Inductive Logic Programming, *The MIT Encyclopedia of the Cognitive Sciences (MITECS)* (MIT Press, Cambridge, 1999)
3. H. Blockeel, L. Dehaspe, B. Demoen, G. Janssens, H. Vandecasteele, Improving the efficiency of inductive logic programming through the use of query packs. J. Artif. Intell. Res. **16**, 135–166 (2002)
4. A. Mutlu, P. Senkul, Y. Kavurucu, Improving the scalability of ILP-based multi-relational concept discovery system through parallelization. Knowl. Based Syst. **24**, 352–368 (2012)
5. V.S. Costa, A. Srinivasan, R. Camacho, H. Blockeel, B. Demoen, G. Janssens, J. Struyf, H. Vandecasteele, W.V. Laer, Query transformations for improving the efficiency of ILP systems. J. Mach. Learn. Res. **4**, 465–491 (2003)
6. A. Srinivasan, A study of two sampling methods for analyzing large datasets with ILP. Data Min. Knowl. Disc. **3**(1), 95–123 (1999)
7. Y. Kavurucu, P. Senkul, I.H. Toroslu, Concept discovery on relational databases: new techniques for search space pruning and rule quality improvement. Knowl. Based Syst. **23**(8), 743–756 (2010)
8. B. Tausend, Representing biases for inductive logic programming,in *ECML. Volume 784 of Lecture Notes in Computer Science*, ed. by F. Bergadano, L.D. Raedt (Springer, Heidelberg, 1994), pp. 427–430
9. L. Dehaspe, L.D. Raedt, Mining association rules in multiple relations, in *ILP. Volume 1297 of Lecture Notes in Computer Science*, ed. by N. Lavrac, S. Dzeroski (Springer, Heidelberg, 1997), pp. 125–132
10. A. Mutlu, P. Senkul, Improving hash table hit ratio of an ilp-based concept discovery system with memoization capabilities, in *ISCIS*, ed. byE. Gelenbe, R. Lent (Springer, Heidelberg, 2012), pp. 261–269
11. D. Maier, *The Theory of Relational Databases* (Computer Science Press, Rockville, 1983)
12. A. Srinivasan, S. Muggleton, R. King, M. Sternberg, Theories for mutagenicity: a study of first-order and feature based induction, Technical report, PRG-TR-8-95 Oxford University Computing Laboratory (1995)
13. A. Srinivasan, R.D. King, S.H. Muggleton, M. Sternberg, The predictive toxicology evaluation challenge, in *IJCAI-97: Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pp. 1–6 (1997)
14. M.J. Pazzani, C. Brunk, G. Silverstein, A knowledge-intensive approach to learning relational concepts, in *Proceedings of the Eighth Intl. Workshop on Machine Learning (ML'91)*, pp. 432–436 (1991)
15. A.K. Akobeng, Understanding diagnostic tests 1: sensitivity, specificity and predictive values. Acta Paediatr. **96**(3), 338–341 (2007)
16. R. Parikh, A. Mathai, S. Parikh, G.C. Sekhar, R. Thomas, Understanding and using sensitivity, specificity and predictive values. Ind. J. Ophthalmol. **56**(1), 45 (2008)

Context Sensitive Search Engine

Remzi Düzağaç and Olcay Taner Yıldız

Abstract In this paper, we use context information extracted from the documents in the collection to improve the performance of the search engine. In first step, we extract context using Lucene, DBPedia-Spotlight, and Wordnet. As the second step, we build a graph using extracted context information. In the third step, in order to group similar contexts, we cluster context graph. In the fourth step, we re-score results using context-clusters and context-information of documents, as well as queries. In the fifth step, we implement a data collection tool to collect gold-standard data. In the sixth and final step, we compare the results of our algorithm with gold-standard data set. According to the experimental results, using context information may improve the search engine performance but the collection should be relatively big.

1 Introduction

Search engines deal with the measurement of how close the source information matches with the user input; thus retrieving the most relevant information to the users. In order to calculate the relevancy, search engines use several parameters such as the popularity of a document, the date of a document, user preferences extracted from usage logs, etc. Therefore, the search engines generally are not able to consider the *meaning* of the input, in other words, it cannot differentiate the context that the search input is related to. This is why the user sometimes might see search results that seem completely out of context comparing to what they searched for. This important problem is the main focus of this paper.

The capability of processing information of the human brain may provide a viable basis for new approaches. In order to process large amount of data, the human brain generates relationships between the current sensory information that represents the external world and previously stored information. This relationship allows human

R. Düzağaç (✉) · O.T. Yıldız (✉)
Işık University, Sile, Istanbul
e-mail: remzi.d@gmail.com

O.T. Yıldız
e-mail: olcaytaner@isikun.edu.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_29

brain to determine actions. This process in the brain provides us an inspiring idea about how the search engines might be improved by adopting the "context over content" approach [11]. This is what this paper contributes to the field of search engines: adding context- based search capability (retrieving documents that do not only contain same words with the search query but also belonging to similar contexts). Instead of focusing on finding the exact content, we take the context from the input information into account and improve the relevancy rate of the results and thus the overall performance of the search.

This paper is organized as follows: We give related work in Sect. 2 and describe the proposed system in Sect. 3. We explain the experimental setup and give results in Sect. 4. Finally, we conclude in Sect. 5.

2 Related Work

In the study of Brown et al. [4], Context Aware Retrieval (CAR) systems and its relationships to information retrieval and filtering are surveyed. They claim that CAR is a sparsely researched area and there are some topics of information retrieval and filtering which is not applied to CAR systems.

Mylonas et al. [13] bring two fields of information retrieval together to present a reliable knowledge-based information extraction method: personalization of the information and retrieval context. The authors state that, this combination provides a unique approach towards the process of measuring relevance, since it introduces the concept of ontological information domain which refers to the enriched representational ground for content meaning, user interests, and contextual conditions.

Li et al. [10] make use of semantic distance for feature-free search query classification. They show advantages of utilizing the number of search results while grouping the web content according to their relevance. In contrast to other machine learning (ML)-based methods, they use a feature-free approach since it's a better fit for ever-changing web data.

In the study of [7], Concept-Based Information Retrieval is performed using explicit semantic analysis. Their concept-based approach (which seems similar to our context over content approach) involves three important contributions: using Explicit Semantic Analysis which refers to real-life concepts resembling human perception, integrating selection phase into concept-based indexing stage, and using three AI-based selection methods for information retrieval.

Atanas Kiryakov et al. [8] introduce an architecture for semantic annotation, indexing and retrieval of documents via semantic resources. Their architecture claims to provide automatic semantic annotation with references to ontology classes and instances. They perform semantic indexing and retrieval where they combine ontology and information retrieval methods.

Schuhmacher and Ponzetto [14] state a knowledge-based approach to web search result clustering and try to solve related research topics focusing on clustering short texts from the web. This information retrieval task is analyzed if it can take advantage

from DBpedia spotlight state-of-the-art entity linking system. The approach uses DBpedia concepts as text seeds to collect topical concept labels. These labels are used to cluster based on their topical similarity. The approach takes web search snippets as inputs and cluster them topically using DBpedia.

3 Proposed System

The system that we designed to implement the context over content approach consists of five different steps: corpus, context extraction, context clustering, labeling, and scoring (Fig. 1). The first step is corpus, which contains preprocessed documents and queries for next steps. The second step is the context extraction. In this part, we use Lucene, DBPedia Spotlight, and Wordnet to extract context information from documents and queries. The third step is the context clustering. We cluster extracted contexts to form groups, which contain similar contexts.Fourth step is labeling. In the labeling step, we use context information extracted from each document and clusters to determine the clusters that each document is related to. Also we label queries in the same manner with documents. Fifth and final step is scoring. In the scoring step, we calculate scores according to the correlations between queries and documents. Brief description of each step is given in the paper. For more detail, see [6].

3.1 Corpus

A corpus is a collection of texts in digital form, over which processes such as retrieval or analysis will performs [5]. In this paper, we use The Westbury Lab Wikipedia Corpus (WLC) [15], as our data source. We apply some preprocessing steps to WLC in order to obtain a corpus which fits our needs. After preprocessing and selection there are 8,000 documents left.

Queries are the questions that are entered into search engine to retrieve results. In order to select queries, first we select n documents ($n \leq 30$) randomly. Then we select a portion of each document randomly and rewrite that portion as a query sentence.

3.2 Context Extraction

The context extraction part of our system, consists of two steps. The first step is analyzing the context with context analyzers. Context analyzers extract terms that represent context information from using three different context sources; Lucene [1], DBPedia-Spotlight [9], and Wordnet [12]. The second step is pairing. After the context extraction step, we pair terms and count pairs in order to find inter-term relevancy. Details of context extraction and pairing steps are explained in [6].

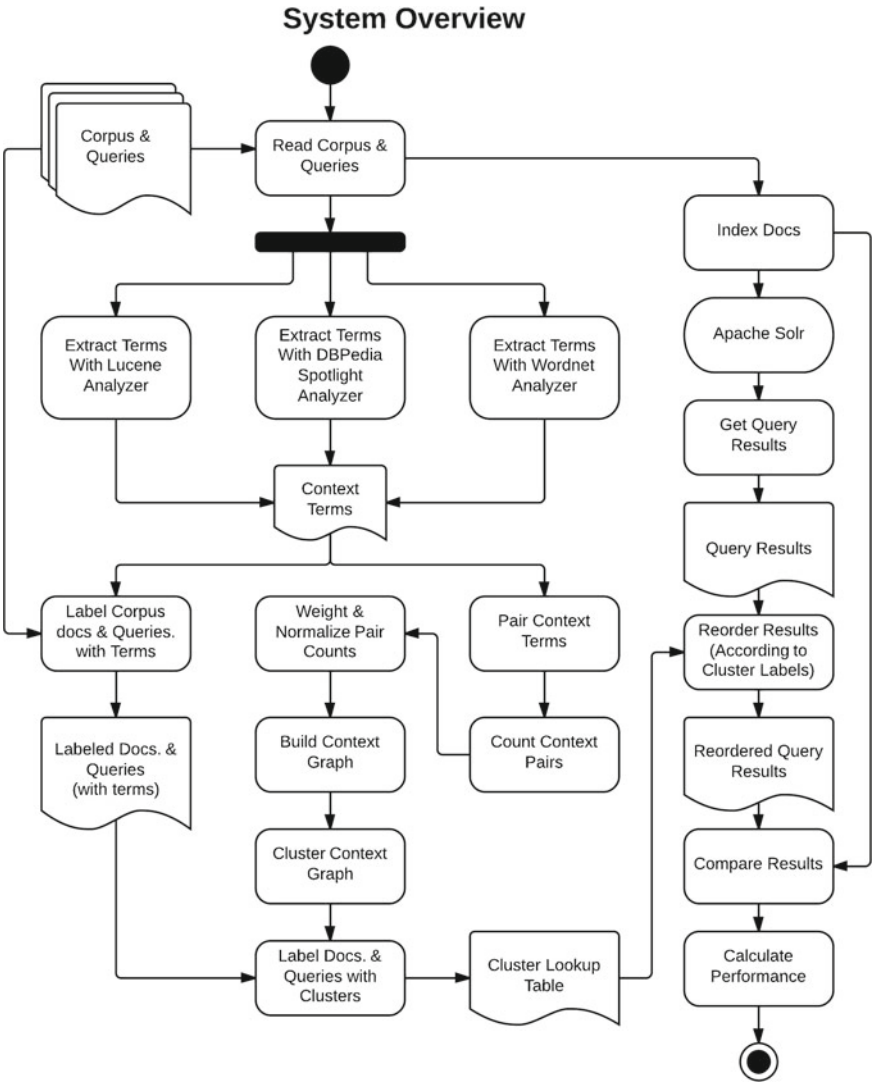


Fig. 1 System overview

3.3 Context Clustering

In order to cluster contexts, first, we build a graph of terms. Terms are connected via distances generated from pair counts that are previously calculated. Details of distance calculation explained in [6].

Second step is clustering graph. In order to cluster our graph of terms, we use Markov Cluster Algorithm (MCL Algorithm) [3]. Finally, we export cluster context-term mapping which shows clusters that contain context terms.

3.4 Labeling and Scoring

In this part of our work, we label documents with clusters, in order to find which document belongs to which cluster. First, we index documents using SOLR [2] and retrieve results for the 30 queries which we have previously selected. Second, we label documents using context information that we extracted in context extraction step. Third, we also label queries using extracted context information. Next, we calculate new scores of documents. In order to calculate new scores we use dot product of cluster-counts of documents and queries. Then we sort results for each query according to new calculated scores.

4 Experiments

4.1 Setup

In order to measure performance, we need reference data to compare re-scored results. Therefore, we have designed and implemented a web application for data collection. The application retrieves top 50 results from SOLR without any extra scoring and shows these results to the user without ordering. The user can categorize the results into four categories. Then the application stores the user selections as scores. Each category has score between zero and three according to its degree of relatedness.

The categorized results of 30 different queries are obtained by our application and are given to several different users. Each query has a set of results for each user. Then we calculate means of scores that are given to each result of query. Then we reorder results according to new scores (Figs. 2, 3, and 4).

4.2 Results

In order to measure performance of our study, we use metrics that frequently used in the literature. These metrics are precision, recall, and f-measure. According to the experimental results, precision of Solr is slightly better. On the other hand, recall of our approach is a little better than Solr. In general, performance for both Solr and our approach are almost same. Precision, recall, and f-measure results also indicate similarities between performances.

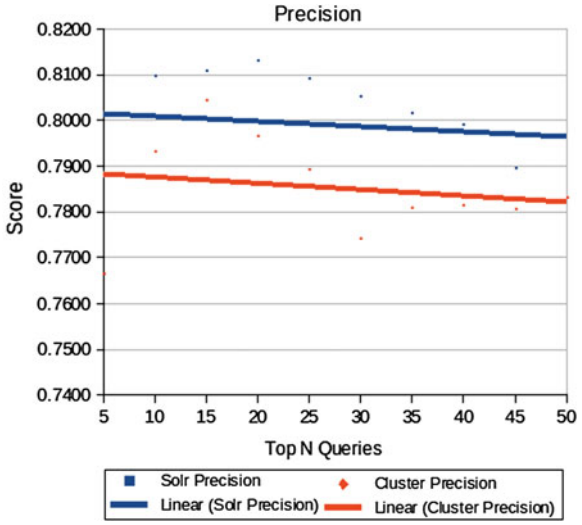


Fig. 2 Precision results of Solr and our approach

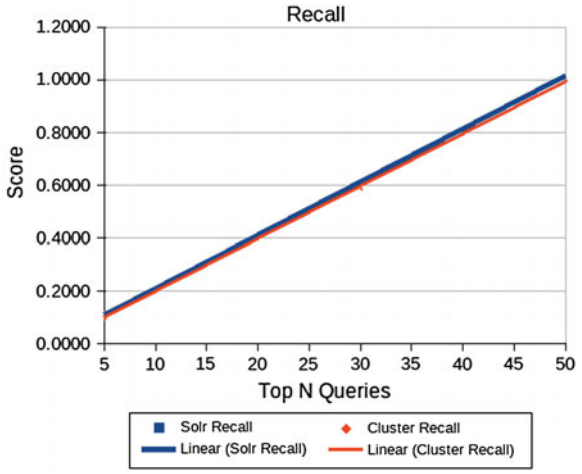


Fig. 3 Recall results of Solr and our approach

The corpus that we use, contains 8,000 documents. This size of corpus is small for a search engine. Using small corpus might be the reason that clusters have not formed good enough to distinguish concepts. Since concepts are not distinguished well enough, re-scoring with clusters do not change the results significantly.

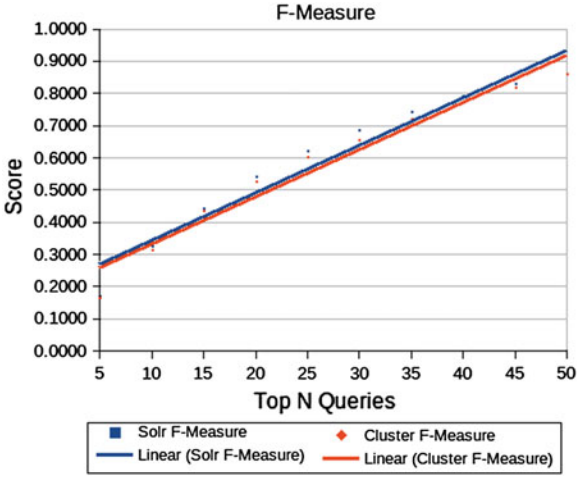


Fig. 4 F-measure results of Solr and our approach

5 Conclusion

In this paper, we focus on retrieving relevant data instead of processing big amounts of data. Llian's context over content approach [11] of human brain has inspired us to develop a method, which will improve search engine performance using a similar way to human thinking process. In our work, we propose a method which uses the context information of documents to improve the search engine performance.

According to the experimental results, we could not improve search engine performance significantly. The reason, mostly, is dependent to our corpus size. Because lack of computational power, we could use 8,000 documents which is very small set for a search engine. Using limited amount of documents is prevented forming clusters well enough. Re-scoring with using non well-formed clusters did not make any significant change.

Our content over context approach, that we presented, forms a basis for our future work. In the future, we plan to implement our design on a distributed system such as Hadoop so that we can process much bigger corpus. We also plan to extend context sources that we used to extract context information.

References

1. Apache lucene. http://lucene.apache.org/core/
2. Apache solr. https://lucene.apache.org/solr/
3. Mcl—a cluster algorithm for graphs. http://micans.org/mcl/
4. P.J. Brown, G.J.F. Jones, Context-aware retrieval: exploring a new environment for information retrieval and information filtering. Personal Ubiquitous Comput. **5**(4), 253–263 (2001)

5. C.D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval* (Cambridge University Press, Cambridge, 2009)
6. R. Duzagac, Improving search engine performance with context extraction using lucene, dbpedia-spotlight, and wordnet. Master's thesis, IK University (2014)
7. O. Egozi, S. Markovitch, E. Gabrilovich, Concept-based information retrieval using explicit semantic analysis. ACM Trans. Inf. Syst. **29**(2), 8:1–8:34 (2011)
8. A. Kiryakov, B. Popov, I. Terziev, D. Manov, D. Ognyanoff, Semantic annotation, indexing, and retrieval. Web Semant. **2**(1), 49–79 (2004)
9. J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, C. Bizer, DBpedia—a large-scale, multilingual knowledge base extracted from wikipedia. Semant. Web J. **26**, 1–58 (2014)
10. L. Li, L. Zhong, G. Xu, M. Kitsuregawa, A feature-free search query classification approach using semantic distance. Expert Syst. Appl. **39**(12), 10,739–10,748 (2012)
11. R.R. Llinas, *I of the Vortex from Neurons to Self* (MIT Press, Cambridge, 2002)
12. G.A. Miller, Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
13. P. Mylonas, D. Vallet, P. Castells, M. FernÁndez, Y. Avrithis, Personalized information retrieval based on context and ontological knowledge. Knowl. Eng. Rev. **23**(1), 73–100 (2008)
14. M. Schuhmacher, S.P. Ponzetto, Exploiting dbpedia for web search results clustering, in *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, AKBC '13* (ACM, New York, 2013), pp. 91–96
15. C. Shaoul, C. Westbury, The westbury lab wikipedia corpus, edmonton, ab: University of alberta. (2010), http://www.psych.ualberta.ca/westburylab/downloads/westburylab.wikicorp.download.html

A Formal Framework for Hypergraph-Based User Profiles

Hilal Tarakci and Nihan Kesim Cicekli

Abstract In this study, we propose a formal framework for user profile representation with hypergraphs. We exploit the framework to aggregate partial profiles of the individual to obtain a complete, multi-domain user model, since we aim to model the user from several perspectives. We use Freebase commons package concepts as predefined domains. The proposed user model is also capable of extracting *user domain capsules*, which models the user for the domain of interest. Moreover, using a hypergraph data structure results in solving connection-based problems easily, since the cost of local operations on a graph is low and independent of the size of the whole graph. Many problems in user modelling domain are connection-based problems, such as recommendation.

Keywords User modelling · User profile · Hypergraph user model

1 Introduction

The popularity of social networking sites has dramatically increased over the last decade. The user's activities on social websites such as his likes, comments, location declarations and friendships reveal important information about his profile. The individual's interests, goals and preferences can be exposed by mining those activities. Social networks differ in nature and are used for different purposes [1]. Therefore, mining separate social networks independently results in partial profiles of the user which merely represents user's interests for one or few domains. In this study,

This work is partially supported by The Scientific and Technical Council of Turkey Grant "TUBITAK EEEAG-112E111".

H. Tarakci (✉)
Department of Computer Engineering, Sakarya University, Sakarya, Turkey
e-mail: htarakci@sakarya.edu.tr

N.K. Cicekli
Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
e-mail: nihan@ceng.metu.edu.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_30

we present a framework to aggregate partial profiles of the individual to obtain a complete, multi-domain user model.

Representing a user profile with graph is a common strategy. The vertices usually represent the items and the users where an edge between a user and an item indicate user's interest on that item. Since the graph is only capable of representing binary relations, other approaches have been proposed for handling higher order relations in user modelling domain. There are a few studies which define user model as bipartite [2] and tripartite graphs [3]. In general, if the number of vertex types n is known in advance and the relations in the user model are binary, an n-partite graph is capable of representing the profile. However, if there are higher-order relations, a hypergraph is more appropriate to represent the user model [4, 5].

In a previous paper, we presented the initial ideas for using hypergraph in the modelling of user profiles [6, 7]. In this paper, the main contribution is a formal framework for hypergraph-based user profiles. It is claimed that aggregating profiles solves the cold-start problem and sparse user model problem [1]. Seamless aggregation of partial user profiles obtained from different knowledge sources is still an unsolved problem. We claim that the proposed hypergraph user model is effective in solving the aggregation of partial profiles.

The paper is organized as follows. Section 2 summarises the related studies. Section 3 formally defines the proposed hypergraph based user model. The application and evaluation details are presented in Sect. 4. Section 5 concludes the paper by summarizing the study.

2 Related Work

In [1], form-based and tag-based profiles are managed separately. The former is a list of attribute-value pairs whereas the latter is a set of weighted tags. The aggregation strategy for form-based profiles is unifying sets of attribute-value pairs. Heterogeneous attribute vocabularies is resolved by using an alignment function, which maps profiles to unified attribute-value space. However, this alignment function may result in duplicate entries in the final user profile. Moreover, when there are conflicts in the aggregated profiles, both values are included in the result. The aggregation of tag-based profiles is accomplished by taking a weighted accumulation of partial tag-based profiles. The authors do not consider aggregating tag-based profiles and form-based profiles with each other. In our paper, we do not make such a distinction. We seamlessly aggregate received partial user profiles by taking their weighted accumulation. We solve heterogeneous vocabulary problem by using Freebase.[1]

In [8], during aggregation the authors address the problem of recurring items and calculating a global weight for them. To achieve this, they keep track of *provenance data* which is the meta data for the user profile item such as the source of the item and the timestamps. This enables the recalculation of item weights during aggregation

[1] Freebase, https://www.freebase.com/.

of the partial profiles. We also keep track of the provenance data by storing the knowledge source, the short term profile date and the exact keyword of the item. We extend this information each time the item and user is bounded together.

3 Data Model and Problem Formulation

A hypergraph is the generalization of an ordinary graph by introducing hyperedges, which are non-empty subsets of the vertex set [9]. Vertices of a hypergraph represents the entities to be modelled such as people and concepts. Hyperedges represent the high-order relations between those entities. Besides hypergraphs, there are property graphs which contains key-value property pairs [10]. In a property graph each node and edge can have multiple key-value pairs whereas in a hypergraph, an edge can connect more than two nodes. Every hypergraph can be represented by a property graph by adding extra key-value pairs to annotate nodes, which are connected by the same hyperedge. For instance, the *Users* hyperedge is represented by assigning the value of the node's *type* as *User* for each user node in the property graph. In this paper, we actually use property graphs, since the graph database we adopted supports property graphs.

In this study, we focus on constructing a holistic user model by aggregating the short term profiles by utilizing the proposed hypergraph data structure. The notations for the proposed hypergraph is summarised in Table 1. Basically the hypergraph user model consists of set of labelled nodes and strongly typed hyperedges. Nodes representing concepts and users are assigned different labels. Similarly, hyperedges responsible for representing the user's interest on an item or indicating the semantic relations between entities belong to different types. On top of these nodes and hyperedges, there are *domains* which divide the hypergraph to overlapping regions to group nodes and hyperedges in the same domain together. In other words, every concept node belongs to one or more domain. In the implementation, we use Freebase commons package as domains and define a *domain starter node* for each domain which connects to the nodes under that domain. The projection of the user in a domain is represented by a *user domain capsule*. The proposed hypergraph facilitates profile aggregation and semantic enhancement with the help of the presented user model structure. A simplified illustration for the hypergraph data structure is presented in (Fig. 1). The figure demonstrates that a user with name *dummyUser* is interested in the item *Pride and Prejudice* which is connected to the *fictional universes* domain. During the *semantic enhancement process*, *Jane Austen* is semantically related to the item with *CreatedBy* relationship. Moreover, the items defining the genre of *Pride and Prejudice* are connected with *HasGenre* relation.

Definition 1 *Partial User Profile*: Partial user profile L_{uts} is the short term profile obtained from the knowledge source s for the user u during time period t. The input of the system are received partial profiles. A partial profile is represented as a vector of terms $[\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n]$.

Table 1 Our hypergraph user model

Notation	Description	Type
L_{uts}	Partial (short term) user profile for user u for a time period t from knowledge source s	A vector of terms
T_s	Uniform time period for receiving short term profiles from source s	Number of days
S	Set of knowledge sources	A set of strings (Facebook, LinkedIn etc.)
U	Set of all registered users	Nodes
U_f	Set of frequent users	Nodes
U_s	Set of semi-frequent users	Nodes
U_r	Set of rare users	Nodes
$f_{\alpha}(u)$	Profile categorization function for user u	A function
$f_{\alpha\text{-div}}(u)$	Function to calculate the diversity of the user profile	A function
$f_{\alpha\text{-den}}(u)$	Function to calculate the density of the user profile	A function
$f_{\alpha\text{-act}}(u)$	Function to calculate the activity of the user profile	A function
Υ_f	The threshold value for being a frequent user	A double
Υ_s	The threshold value for being a semi-frequent user	A double
Υ_{div}	The threshold value for diversity	A double
Υ_{den}	The threshold value for density	A double
C	Set of entities (concepts)	Nodes
$D_{[d]}$	Domain starter node for each predefined domain d	Nodes
E_{bind}	Metadata for user-item (interest) relation	A hyperedge
E_{inner}	The semantic relation between items (entities and named entities)	A hyperedge
Υ_{inner}	The semantic relation threshold which defines the enhance limit	An integer
E_{domain}	The domain bind between domain starter node and items	A hyperedge
Υ_{domain}	Domain threshold value to decide the number of the domain connections to represent	An integer
$f_{\text{ud}}(u, d)$	User domain capsule function	A function
f_{decay}	Profile decay function	A function
$f_{\text{sim}}(c, u, d)$	Similarity function for concept and user domain profile	A function
$f_{\text{userSim}}(u_1, u_2, d)$	Similarity function for users	A function
P_{u}	General (long term) user profile	A sub hypergraph
SemEnh	Algorithm for semantic enhancement	An algorithm
ProfAgg	Algorithm for profile aggregation	An algorithm

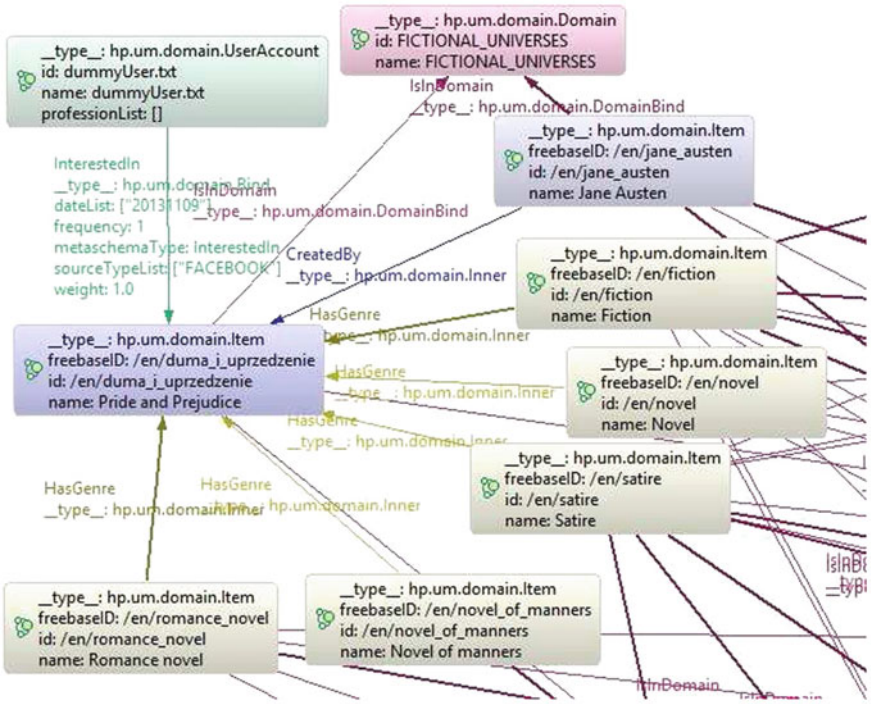


Fig. 1 Hypergraph data structure illustration

User Categorization: People have different social web usage habits. A user may be frequently active in social websites, whereas another may scarcely use his accounts. Even two frequent social web users may show differences in their usage behaviour. A user's actions may show broad interest in many domains whereas another may exhibit deep interest in few domains. Categorizing users according to their usage habits enables definition of ad hoc algorithms for each user type. Let U denote the set of all registered users. U consists of the union of frequent users U_{f}, semi-frequent users U_s and rare users U_r. Frequent users have well-defined profiles for probably many domains whereas semi-frequent users have defined profiles for few domains. Rare users consist of new users and users who barely use their social web accounts. The category to which the user belongs may change in time according to a *profile categorization function* $f_\alpha(u)$ and two threshold values. $f_\alpha(u)$ is calculated as a weighted combination of three sub-functions: $f_\alpha(u) = x.f_{\alpha\text{-div}}(u) + y.f_{\alpha\text{-den}}(u) + z.f_{\alpha\text{-act}}(u)$ where x, y and z are non-negative impact factors and their sum is equal to 1. $f_{\alpha\text{-div}}(u)$ calculates the diversity of profile amongst several domains, $f_{\alpha\text{-den}}(u)$ the density of profile under a specific domain and $f_{\alpha\text{-act}}(u)$ the activity degree on the social web accounts of the user. $f_{\alpha\text{-div}}(u)$ computes the diversification of the user's profile over domains by calculating the number of domains the user have items more than a threshold Υ_{div}. Namely, users who have items distributed in many domains

have high $f_{\alpha\text{-div}}(u)$ values. $f_{\alpha\text{-den}}(u)$ computes the deepness of the user's profile in one particular domain. It is computed by calculating the number of domains user have items more than a threshold Υ_{den} where $\Upsilon_{\text{den}} > \Upsilon_{\text{div}}$. In other words, $f_{\alpha\text{-den}}(u)$ value is high for users whose profiles are defined in detail for a number of domains. $f_{\alpha\text{-act}}(u)$ computes the recent update rate of the user. It is calculated when the latest short-term profile for the user is received. The score is based on the number of modifications and extensions applied to the original user model. When the value of the profile categorization function $f_{\alpha}(u)$ is above a threshold Υ_{f}, the user is classified as a frequent user. If the score is between Υ_{f} and $\Upsilon_s < \Upsilon_{\text{f}}$, the user is a semi-frequent user. Otherwise, the user is categorised as a rare user.

Domains: Our proposed hypergraph aims to model the user from several perspectives. In order to achieve this, we use Freebase commons package concepts as predefined domains. In fact, Freebase also introduces these concepts as domains on its home page. Domains are represented with separate *domain starter nodes*. Let $D[d]$ denotes the *domain starter node* for the domain with name d. For instance, D_{tennis}, D_{sports}, $D_{\text{fictional-universes}}$ represents starter nodes for domains *tennis*, *sports* and *fictional universes* domains, respectively. The domains may overlap with each other. This situation does not lead to a problem, since we handle each domain as a separate projection of the user's profile.

Definition 2 *User Domain Capsule*: User domain capsule of the user u for the domain of interest d is the sub hypergraph which *maximally covers* the user under the domain d. The proposed user domain capsule resembles the *news capsule* presented in [4], which is constructed by partitioning the hypergraph into a predefined number of sub-graphs. *News capsules* are not per-user in order to enable inference on the graph for other users. In our study, we use the *capsule* notion in a different way, to obtain a compact structure to capture the user's profile for a particular domain. To obtain *user domain capsules*, the item nodes which are connected to the domain and reachable from the user are collected. When the user u is connected with an item $c \in C$, the provenance data should be kept to use the item's history in the weight calculation algorithm. The weight calculation algorithm computes the interest of the user on an item by considering the provenance data. For instance, as the time passes, the weight of the interest decays. E_{bind} hyperedge type is used to keep the provenance data. For the relations between concepts E_{inner} hyperedge type is used. The relation type between the concepts in Freebase under the domain of concern is kept in the property *freebase Relation*. We used a subset of *Freebase metaschema properties* to model the semantics between the concepts.

Weight and Similarity: The *user domain capsule* of the user u for the domain of concern d is calculated by a function $f_{\text{ud}}(u, d)$. The function returns a vector of concept-weight pairs which represents the user's projection on the domain of interest in vector space model. f_{decay} function ensures that the weight of the most recently created or updated concept is supported more than older profile items. In order to decide whether the user is interested in a concept, the similarity between the concept and user profile is calculated according to the selected similarity metric

[11, 13]. There are similarity calculation approaches including measuring semantic similarity between words using web documents [11, 12]. In this study, we define a similarity function $f_{\text{sim}}(c, u, d)$ which considers both similarity and semantic relatedness. The function moves the user profile to vector space model by obtaining the *user domain capsule* using $f_{\text{ud}}(u, d)$ and calculates the similarity score based on the cosine similarity between the concept c and the concepts in the domain user profile. In order to compute the similarity of two users under a domain, we define $f_{\text{userSim}}(u_1, u_2, d)$ which takes the user domain capsule that has fewer concepts as pivot and calculate $f_{\text{sim}}(c, u, d)$ score for each c in the pivot user domain capsule and make a weighted accumulation of the highest, lowest and average similarity scores.

Profile Aggregation: We receive short term profiles for users on a regular basis. To obtain a complete multi-domain profile of the user, short-term profiles are aggregated by using the *profile aggregation function* $f_p(u, L_{\text{uts}})$.

Definition 3 *User Profile*: The user profile P_{u} is the aggregated user model for the user u and it is the hypergraph which consists of the user u, the interest nodes of u and the hyperedges between them. $f_p(u, L_{\text{uts}})$ takes the short term profile of the user as input and outputs the general user profile denoted as P_{u}. *Profile aggregation function* aggregates the short term profile by the following algorithm:

```
foreach term t in L_uts:
 disambiguate term t from knowledge base.
 if the item is already in the hypergraph:
    if the item is already connected to the user:
    update provenance data.
    else: create a bind between the user and the item.
 else: create the item and connect it with the user.
       decide domains for the item from knowledge base.
       connect the item to the domain starter nodes of its
       domains.
       enhance the item by using the middle ontology.
       foreach enhancing item:
          create node, decide domains.
          connect the item with the enhancing item semantically.
retrieve the user and reachable item nodes, output P_u
```

4 Application and Evaluation of Formal Framework

The initial dataset is prepared by collecting short term profiles from Facebook accounts of 204 users during two months by mining page likes. 12 short term profile sets are constructed by taking the time period as 3,4 or 7 days. Since the number of users is small, user categorization is not applied and concepts and named concepts are not discriminated. During evaluation, each user is extracted from the dataset and

the hypergraph is populated with the remaining user. Afterwards, during aggregation of the user to the prevously populated hypergraph, when the item is already in the graph, this is considered as a hit. For 204 users, the average of hits-to-total items ratio is calculated as *0.61*. In the baseline, the knowledge base usage and enhancement is removed and the same data is evaluated. The average hits-to-total ratio for the baseline is *0.25*. The resulting scores show that usage of a knowledge base and the enhancement procedure successfully predicts the user's future interests. The dataset is prepared by collecting only page likes; using other social activities may result in more accurate short term profiles. We are going to improve our dataset by collecting users from public feeds of social websites and analyse them for a longer period. Furthermore, we are going to accomplish more detailed and comprehensive evaluations.

5 Conclusions

In this paper, we presented a formal framework for managing a hypergraph user model. We enabled seamless aggregation of partial user profiles with the help of the semantic enhancement of short term profile items. During semantic enhancement, the short-term profile terms become semantic nodes in the graph and the item nodes are attached to their domains and other related items with specialised hyperedges. Usage of domains enable extraction of *user domain capsules*, which are domain projections of users' profiles. Moreover, a number of user modelling domain problems are connected-data problems which could be solved easily by using a graph data structure. As future work, we are going to evaluate the framework against a bigger dataset and implement and evaluate a recommendation case study, which uses the proposed system.

References

1. F. Abel, E. Herder, G.J. Houben, N. Henze, D. Krause, Cross-system user modeling and personalization on the social web. User Model. User-Adapt. Interact. **23**(2–3), 169–209 (2013)
2. A. Tiroshi, S. Berkovsky, M.A. Kaafar, T. Chen, T. Kuflik, Cross social networks interests predictions based ongraph features, in *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)* (ACM, New York, 2013), pp. 319–322
3. B. Chen, J. Wang, Q. Huang, T. Mei, Personalized video recommendation through tripartite graph propagation. in *Proceedings of the 20th ACM International Conference on Multimedia* (ACM, 2012), pp. 1133–1136
4. L. Li, T. Li, News recommendation via hypergraph learning: encapsulation of user behavior and news content. in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining* (ACM, 2013), pp. 305–314
5. T. Kramr, M. Barla, M. Bielikov, Personalizing search using socially enhanced interest model, built from the stream of user's activity. J. Web Eng. **12**(1–2), 65–92 (2013)

6. H. Tarakci, N.K. Cicekli, Ubiquitous fuzzy user modeling for multi-application environments by mining socially enhanced online traces. in *User Modeling, Adaptation, and Personalization* (Springer, Berlin Heidelberg, 2012), pp. 387–390
7. H. Taraki, N.K. Cicekli, UCASFUM: A Ubiquitous context-aware semantic fuzzy user modeling system. In *KEOD* (2012), pp. 278–283
8. F. Orlandi, J. Breslin, A. Passant, Aggregated, interoperable and multi-domain user profiles for the social web. in *Proceedings of the 8th International Conference on Semantic Systems* (ACM, 2012), pp. 41–48
9. G. Gallo, G. Longo, S. Pallottino, S. Nguyen, Directed hypergraphs and applications. Discret. Appl. Math. **42**(2), 177–201 (1993)
10. I. Robinson, J. Webber, E. Eifrem, *Graph Databases* (O'Reilly Media Inc., Sebastopol, 2013)
11. S.A. Takale, S. Nandgaonkar, Measuring semantic similarity between words using web documents. Int. J. Adv. Comput. Sci. Appl. IJACSA **1**(4), 78–85 (2010)
12. L. Zhiqiang, S. Werimin, Y. Zhenhua, Measuring semantic similarity between words using wikipedia. in *Web Information Systems and Mining, 2009. WISM 2009. International Conference on* (IEEE, 2009), pp. 251–255
13. P. Ilakiya, M. Sumathi, S. Karthik, A survey on semantic similarity between words in semantic web. in *Radar, Communication and Computing (ICRCC), 2012 International Conference on* (IEEE, 2012), pp. 213–216

A Survey of Data Stream Processing Tools

Marcin Gorawski, Anna Gorawska and Krzysztof Pasterak

Abstract In current international context boundaries set for applications are being pushed by the emergence of bursty and time-varying data streams required to be processed in near real-time. Furthermore, traditional techniques for data mining cannot be applied to data streams. Thus, stream-based applications must exhibit to excel at a plurality of requirements. According to defined rules presented in previous promulgated researches on this subject we differ stream-based applications and evaluate their aptitude to stream sources management. By this work we intend to present features and drawbacks of existing software coming from both industry and academic world, along with outlining our contribution to this field.

Keywords Data mining · Data stream processing · Real-time processing · Tool comparison

1 Introduction

A large class of applications has been distinguished in accordance to source data being generated asynchronously in an unpredictable manner. Data streams [13, 15], as they were called, are massive sequences of data that are rapid, unbounded in size, real-time and very often contain multidimensional data items, i.e., tuples. As it was outlined in [15] there are attempts to adjust classical systems as DBMS or rule engines to manage data stream processing. However, they tend to fail in terms of the most important real-time processing requirements. Consequently, applications oriented

M. Gorawski (✉) · A. Gorawska · K. Pasterak
Institute of Computer Science, Silesian University of Technology,
Akademicka 16,
44–100 Gliwice, Poland
e-mail: Marcin.Gorawski@polsl.pl

A. Gorawska
e-mail: Anna.Gorawska@polsl.pl

K. Pasterak
e-mail: Krzysztof.Pasterak@polsl.pl

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_31

toward processing data streams are substantially different from conventional ones, thereby ill-equipped, which cannot meet high-volume and low-latency processing criterions. Tools primarily designed to serve this purpose are far more effective and reliable in managing stream-oriented workloads.

1.1 Research Criterions

While creating data stream processing system from a scratch, following real-time processing requirements [15] ought to be addressed:

1. *In-stream processing*. Where system processes data without any additional memory operations, i.e., storing, in a non-polling processing model. While providing for 'straight-through' operations, time-intensive storage operations are omitted in processing pipeline which results in increasing system's time efficiency.
2. *Stream SQL-like language*. Built-in dedicated language where primitives and operators express continuous stream processing characteristics.
3. *Handling streams imperfections*. With handling unpredictable sources resiliency is essential in terms of performing calculation on partial, deficient, delayed, or out-of-order data.
4. *Predictable outcomes*. Processing pipeline should secure time-ordered processing of data, so when reprocessing sequence of tuples due to failure occurrence, outcome would be the same as if failover did not happen. Therefore, performing operations on stream data ought to lead to predictable, deterministic and repeatable results.
5. *Integration of current and historical data*. Although data stream processing model aims at management of current real-time data, it must also provide basis for historical data processing. By extending systems capabilities in such manner we gain possibility to identify whether data pattern observed in current online data is consistent with already established ones,
6. *Safety and availability*. Failure occurrence is a critical concern. Thus, stream-based applications have to be up and in case of failover seamlessly recover ensuring data availability and integrity.
7. *Distributed processing*. Spreading processing across multiple computing units is sufficient in terms of gaining scalability.
8. *Minimization of response time*. Even if all previous features characterize system when it is not optimized towards minimizing overheads it will not be able to react efficiently to continual inputs from various stream sources. Therefore, when escalation in number of data volumes in time is observed, system must procure responses without latency in a continuous timely fashion.

Sensorization of the world have posed many challenges gathered as a list of eight requirements. Next two sections will present applications which target those issues. In Sect. 2 five selected data stream processing systems will be shortly described, while

Sect. 3 will serve a purpose of propagating our contribution in a form of constantly maturing systems. Finally, we summarize features of described systems and provide concluding remarks.

2 Selective Overview of Stream-Based Processing Systems

2.1 Aurora and Borealis

In its early stage Aurora [4] system was built as a single site stream-based application at Brown University. Thus, it is not prepared for distributed and parallel processing. Unfortunately, it does not satisfy requirements of scalability, reliability, and data safety. Borealis processing engine [3] is a successor to the previously mentioned Aurora system. Among modules of the Borealis system there are: stream processing engine, load manager, and load shedder. Further, a mechanism of fault tolerance satisfies the need to provide safety and availability in case of system's failure, while revision processing mechanism handles tuple's imperfections by correcting erroneous ones. Apart from mentioned features it incorporates textual query definition language built on the XML standard, dynamic revision of query results, and query modification.

2.2 Apache Storm

Next system is an open-source and free product called Storm [2], which focus is set on real-time data processing with ability to distribute and parallelize computation. Moreover, the Storm cater fault tolerance in case of tuple imperfections and ensures processing even if failover occurred. With an architecture inspired by Hadoop, the Apache Storm is easy to use and what is more important it can be easily linked with most of existing queuing and database technologies. Previously mentioned systems came from academic world.

2.3 Apache Samza

Another stream-based application from Apache [1] aims at performing computation over data streams by combining Apache Kafka's messaging and Apache Hadoop NextGen MapReduce (YARN). The last ones purpose is provision of fault tolerance, processor isolation, security, and resource management. Input streams of events are decomposed and partitioned so that data flow graph is created. Each graph contains multiple streams and jobs (i.e., processing primitives), which describe different utilities of continuous queries.

2.4 Microsoft StreamInsight

Microsoft StreamInsight [5] is an extensible framework, merged with Microsoft SQL Server, which leverages continuous processing of long-running (in theory even infinite in time) streams of events. This system has been architected on the basis of CEDR [7] project. The most important aspect is ability to integrate Microsoft StreamInsight with almost any field of interest by constructing a proper processing pipeline. Logic of queries is based on a temporal time model and operator algebra. Meanwhile, the system aims at revealing patterns and trends.

2.5 StreamGlobe

StreamGlobe [14] is a grid-based P2P DSMS. With stream and result sharing mechanisms as well as early filtering and aggregation it enables to avoid redundancy in terms of data stream transmissions and computation. Moreover, by using StreamGlobe reduction of network traffic and pear load is observable.

3 Our Contribution to Data Stream Processing

In the previous section five selected systems where described shortly. In accordance to mentioned features we want to present our contribution to this field.

3.1 StreamAPAS

The first system, StreamAPAS [10, 11], implements following stream operators: selection, projection, duplicate elimination, grouping (aggregation), join, union, intersection, and difference. In addition, it contains also a stream query language, as well as a query graph optimizer.

The first important feature of the StreamAPAS system is time model—*mixed* time model [10], combining both temporal and negative [9] time models. The mixed time model was introduced in order to eliminate disadvantages of the aforementioned models.

The stream processing system StreamAPAS includes a declarative stream query language [11]. User defines format of final outcome rather than declaring subsequent steps of the algorithm (as in procedural languages). Therefore, process of defining stream query is independent of the physical architecture of the system. Moreover, the StreamAPAS language contains some features that are present in object oriented languages, which increases its functionality extension capabilities.

Another advantage is existence of attribute trees, i.e., hierarchical structures used instead of simple values for defining tuple's data, which makes organizing and managing easier.

The next important feature of the StreamAPAS system is the usage of operator partitions in a query graph optimization process. The operator's partition is also called the compound operator, which includes variety of simple (regular) stream operators. The operator partition is seen by the system as an usual operator and its internal components exchange tuples without buffering. Such objects are created when the total time or memory characteristic of certain simple operators are worse than characteristic of them bounded in one operator's partition.

3.2 THSPS

Next step in our research on the stream-based processing systems was creation of yet another system, THSPS, which supports selection, transformations (projection and mathematical operations), join, and aggregation operations. In this system our focus was set on different time model, the *tri-temporal time model* like in Microsoft StreamInsight, where each tuple time is described by three dimensions:

- *occurrence time*. Time interval when the event described by tuple has appeared,
- *validity time*. Time interval in which tuple can be analyzed by the system,
- *system time*. Timestamp representing tuple's arrival time, i.e., moment in time, when it has arrived to the system.

Among many advantages of the tri-temporal time model the main benefit is possibility of analyzing both current and historical events—which enhances views of stream sources.

Each tuple represents a real-world event, which is described by two time intervals, while the system time is used rather by the system for maintaining proper order of tuples and is not directly connected with the event itself. Therefore, operators that use multiple tuples in one cycle (i.e., join and aggregation operators) need to consider these time dimensions during their operations. When joining data is concerned, outcome tuple is characterized by both time dimensions narrowed to the common part of the corresponding dimensions of two joined tuples. Thus, the result of joining operation is an event that occurs when two input tuples overlap.

Analogical situation occurs when in processing pipeline aggregation operator is used. Input sequence of tuples after aggregation is merged into one. Outcome tuple consists of aggregated value (e.g., sum, mean value) and time intervals equal to sum of the corresponding time dimensions from all aggregated tuples. So, resulting tuple carries information about an event which lasts as long as all aggregated tuples.

Another sufficient module of the THSPS system is the query graph optimizer, which rebuilds internal structure of the graph to reduce number of unnecessarily processed tuples. This graph represents processing pipeline. For example, moving

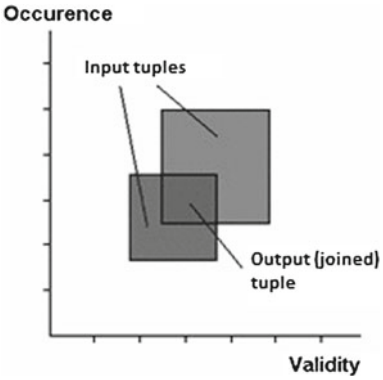


Fig. 1 Join operation in tri-temporal time model

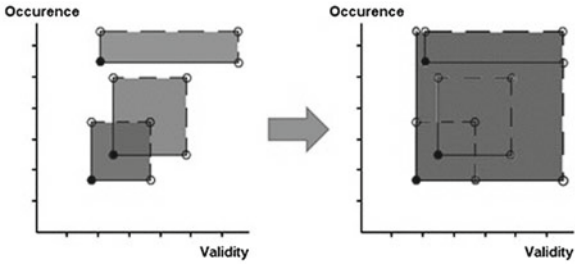


Fig. 2 Aggregation operation in tri-temporal time model

a selection before other operators does not change query's result. Moreover, it optimizes query by reducing number of tuples by eliminating those which do not satisfy selection operator's predicate. Thus, tuples will be discarded on the very beginning of their path through the graph. The major rule of the query optimizer is that operators that have the smallest *selectivity in time* (i.e., the difference between the number of input and output tuples divided by the operator life time), should be considered as candidates to transferring to the beginning of the graph.

The other feature of the THSPS system is existence of *joined tuple decorating mechanism*. Its principle is to create a tuple decorator, i.e., an object which has the same interface as a regular tuple, but holds inside two physical tuples that have been joined. Such concept was designed to reduce total number of tuples currently stored in the system, as well as to increase the speed of producing results in join operator—due to decreasing memory allocation and attribute copying operations.

3.3 AGKPStream

After collecting experiences from both previously created systems in the AGKP-Stream system [13] temporal time model was chosen as well as following stream operators: selection, projection, union, and join (in three variants: *cross-join*, *equi-join* and *theta-join*). Each operator defines two basic transformations aiming at schema or attribute modifications and reformations.

In the AGKPStream system, the temporal time model was selected. Each tuple denotes a time-lasting event that is represented by the time interval called *tuple life time* and is described by two border timestamps: beginning and end. While the tuple is valid (the current system time lays between these two timestamps) it can be processed.

All operators in the AGKPStream system form and DAG likewise in the Aurora system, each representing a single continuous query. The work of every processing primitive is managed by the scheduler, which can work in one of following modes: query or global level.

Global scheduler chooses one query each time and calls it to work, by activating its local scheduler. Query's schedulers manages operators by pointing which should work in a particular moment in time, basing on various factors.

Similarly to the THSPS system, the AGKPStream system uses joined tuple decorator mechanism. In both solutions, such mechanism enables the system to join only two tuples. However, in some special cases, e.g., when multiple join operators are connected in cascade, resulting joined tuples (when using decorating mechanism) are also formed in the same manner—subsequent decorating objects points other decorators etc.

4 Evaluation of Data Stream Processing Systems

While conducting presented research we have examined tools supportable over data stream processing paradigm. Three out of eight were created by our team during past years, while remaining five were taken as a reference point. Each system was examined in terms of requirements as follows: in-stream processing (1), stream language (2), handling stream imperfections (3), predictable outcomes (4), data safety and availability (5), integration of stored and stream data (6), distribution and scalability (7), and instantaneous outcome (8).

Table 1 presents eight requirements listed in the introduction versus their coverage in aforementioned systems, where each entry has one of five values:

- *yes*. System natively provides this capability,
- *no*. System does not support this feature,
- *possible*. This challenge was not solved in the system yet, but it is possible to do so,

- *hard*. Theoretically adaptation to this requirement is possible, but it would require substantial changes in the system,
- ? Lack of information.

Table 1 Comparison of described systems with emphasis to the eight requirements

System	Req. 1	Req. 2	Req. 3	Req. 4	Req. 5	Req. 6	Req. 7	Req. 8
Borealis	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Apache Storm	Yes	No	Yes	Yes	?	Yes	Yes	Yes
Apache Samza	Yes	No	Yes	Yes	Possible	Yes	Yes	Yes
StreamInsight	Yes	No	Yes	Yes	?	Yes	Yes	Yes
StreamGlobe	Yes	?	Possible	Yes	?	?	Yes	Yes
StreamAPAS	Yes	Yes	Yes	Yes	Hard	Yes	Yes	Yes
THSPS	Yes	Possible	Hard	Hard	Hard	Possible	No	Yes
AGKPStream	Yes	Possible	Yes	Yes	Hard	Possible	Yes	Yes

As Table 1 shows, all tested systems fulfilled condition to yield responses in a timely manner with emphasis to processing data in the 'straight-through' model. Another fundamental rules from 3 and 4 were satisfied by most of the systems, while integrating historical with real-time data and possession of native stream querying language were arised as the most problematic requirements.

5 Concluding Remarks

The aim of this paper was to supply with a selective survey of tools enabling data stream processing. Created comparative review was based on assumption that mentioned eight requirements present data stream processing model's characteristics adequately. In previous sections we have supplied tabular results of conducted research where goal was to provide user with an upfront knowledge what types of systems are available on the market and how they rise to streaming challenges.

Unfortunately, Aurora, Borealis, and Stream [6] projects are no longer active but it is not an indication of meaninglessness of research continuation. On the contrary, there are still numerous of new and subtle problems unsolved thereby our ongoing work on the topic is even more valuable. We are not only pursuing evolution of data stream processing systems, but also joining its most important objectives with other areas of our domain expertise, i.e., databases and data warehouses. With this combination first attempts to create a Stream Data Warehouse were made [12]. Moreover, many researches all over the world focus their efforts on adapting data stream processing model and big data to create new solution in presented domain, like StreamGlobe, PIPES [8], Apache Samza, and Storm, among with many other attempts.

References

1. Apache Samza official website. http://samza.incubator.apache.org/
2. Apache Storm official website. http://storm.incubator.apache.org/
3. D.J. Abadi, Y. Ahmad, M. Balazinska, U. Çetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, S.B. Zdonik. The design of the borealis stream processing engine. in: *CIDR*, pp. 277–289, 2005
4. D.J. Abadi, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, S. Zdonik, Aurora: a new model and architecture for data stream management. VLDB J. **12**(2), 120–139 (2003)
5. M. Ali, B. Chandramouli, J. Goldstein, and R. Schindlauer. The extensibility framework in microsoft streaminsight. in: *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, ICDE '11, IEEE Computer Society*, pp. 1242–1253. 2011
6. A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, M. Datar, K. Ito, R. Motwani, U. Srivastava, J. Widom, Stream: the stanford stream data manager. Technical Report 2003–21, Stanford InfoLab, 2003
7. R.S. Barga, J. Goldstein, M.H. Ali, M. Hong. Consistent streaming through time: a vision for event stream processing. in: *CIDR*, pp. 363–374. www.cidrdb.org
8. M. Cammert, C. Heinz, J. Krmer, A. Markowetz, B. Seeger, Pipes: a multi-threaded publish-subscribe architecture for continuous queries over streaming data sources. Technical report, 2003
9. T.M. Ghanem, M.A. Hammad, M.F. Mokbel, W.G. Aref, A.K. Elmagarmid, Incremental evaluation of sliding-window queries over data streams. IEEE Trans. Knowl. Data Eng. **19**(1), 57–72 (2007)
10. M. Gorawski, A. Chrószcz, Query processing using negative and temporal tuples in stream query engines. in: *Advances in Software Engineering Techniques—4th IFIP TC 2 Central and East European Conference on Software Engineering Techniques, CEE-SET 2009, Revised Selected Papers, volume 7054 of Lecture Notes in Computer Science*, (Springer, 2009) pp. 70–83
11. M. Gorawski, A. Chrószcz, StreamAPAS: query language and data model. in: *2009 International Conference on Complex, Intelligent and Software Intensive Systems, CISIS 2009, IEEE Computer Society*, pp. 75–82, 2009
12. M. Gorawski, A. Gorawska, Research on the Stream ETL Process. in: *Beyond Databases, Architectures, and Structures, volume 424 of Communications in Computer and Information Science*, (Springer, 2014) pp. 61–71
13. M. Gorawski, A. Gorawska, K. Pasterak, Evaluation and development perspectives of stream data processing systems, in: *Computer Networks*, vol. 370 (Springer, Berlin, 2013), pp. 300–311
14. R. Kuntschke, B. Stegmaier, A. Kemper, A. Reiser, Streamglobe: processing and sharing data streams in grid-based p2p infrastructures. in: *Proceedings of the 31st International Conference on Very Large Data Bases, VLDB*, pp. 1259–1262. ACM, 2005
15. M. Stonebraker, U. Çetintemel, S. Zdonik, The 8 requirements of real-time stream processing. SIGMOD Rec. **34**(4), 42–47 (2005)

Distributed RDFS Reasoning with MapReduce

Yigit Cetin and Osman Abul

Abstract We live in big data age in which many computational tasks either generate or need to use large datasets. This makes parallel and distributed computing a key for scalability. MapReduce is a programming model for processing large datasets in parallel and distributed fashion on cluster of computers. Today, since the size and complexity of RDFS documents increase rapidly, RDFS reasoning problem has to embrace and address the big data solutions. The output of RDFS reasoning job can be input to another job and the output of RDFS reasoning jobs grow big as the input documents gets bigger. In this study, an indexing method is proposed to speed up the RDFS reasoning over Hadoop clusters. We also explore the utility of caching and Hadoop ecosystem tools Apache Hive and Apache Pig for this task. Experimental evaluations on Dbpedia and Freebase datasets show that the indexing method is quite effective and offers scalable solutions. Performance of caching and Apache Hive is found acceptable too.

Keywords Big data · Mapreduce · Hadoop · Rdfs reasoning

1 Introduction

Semantic web is an Internet technology in which documents can be processed and interpreted by software systems [1]. The overall objective is to make data stored in the web machine understandable/reasonable without human intervention. RDF/RDFS are languages developed to address the semantic data standards. Hence, RDFS

Y. Cetin · O. Abul (✉)
Department of Computer Engineering, TOBB University of Economics and Technology, Ankara, Turkey
e-mail: osmanabul@etu.edu.tr

Y. Cetin
e-mail: ycetin@etu.edu.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_32

language enables reasoning from semantically defined data. In its nutshell, RDFS documents contain facts (extensional database) expressed as triples and rules (intentional database) to derive new triples. If RDFS document content (dataset) is really huge, RDFS reasoning process may spend too much time to generate derived triples. Moreover, disk IO performance may become bottleneck for performance requirements, the reason is simply the frequent and large number of disk accesses during the operation. Parallel processing [2] and cloud computing are good choices to overcome these obstacles [3].

The Apache Hadoop software library is a framework that allows one to distribute and process large data sets across clusters of computers using simple programming models [4]. It supports parallel processing with MapReduce approach [5] and due to its design increases the performance of computing. Apache Hive [6] and Apache Pig [7] are important facilities in Hadoop ecosystem. Apache Hive is a data warehouse infrastructure built on top of the Hadoop core and facilitates analyzing, querying and managing large datasets [8]. Apache Pig is a platform to analyze large data sets in a pipeline manner. As a result, MapReduce core and the Hadoop facilities are among the good candidates for high performance RDFS Reasoning. MapReduce-based RDFS processing has also been extended to high-performance query processing [9] and building scalable storages [10].

In this work, we propose to use indexing for RDFS reasoning. The method simply indexes triples in Apache Solr engine and develops MapReduce algorithms on Hadoop. We also explore the utility of simple caching method, Apache Hive and Apache Pig as alternatives on Hadoop. The experimental results on two datasets (Dbpedia and Freebase) suggest that indexing, caching, and Apache Hive are viable choices. Our study is the first in the literature addressing the RDFS reasoning over Hadoop clusters using different approaches. This is the main contribution of the work.

The rest of this paper is organized as follows: after giving necessary background in the rest of this section, we detail Hadoop RDFS reasoning methods in Sect. 2, then Sect. 3 presents our experimental evaluation. Finally Sect. 4 concludes.

1.1 RDFS Reasoning

RDFS is an extension of RDF that enables users to define the vocabulary used in RDF documents [11]. RDFS language allows definition of special relations between resources. Here is an example of two triples in the form of subject-predicate-object:

Footballer rdfs:subClassOf Person.
Ronaldo rdf:type Footballer.

The first construct (predicate) rdfs:subclassOf is a special relation to define relation between two resources: Footballer and Person. The second predicate rdf:type defines the relation between Ronaldo and Footballer resources. These kind of relations are not context dependent, i.e., every dataset has the same meaning for these relations. Exploiting the meanings we can derive new triples and this is indeed what the RDFS

Table 1 RDFS reasoning rules

Number	Input triple (s)	Derived triple
1	s p o (if o is a literal)	:n rdf:type rdfs:Literal
2	p rdfs:domain x and s p o	s rdf:type x
3	p rdfs:range x and s p o	o rdf:type x
4a	s p o	s rdf:type rdfs:Resource
4b	s p o	o rdf:type rdfs:Resource
5	p rdfs:subPropertyOf q and q rdfs:subPropertyOf r	p rdfs:subPropertyOf r
6	p rdf:type rdf:Property	p rdfs:subPropertyOf p
7	s p o and p rdfs:subPropertyOf q	s q o
8	s rdf:type rdfs:Class	s rdfs:subClassOf rdfs:Resource
9	s rdf:type x and x rdfs:subClassOf y	s rdf:type y
10	s rdf:type rdfs:Class	s rdfs:subClassOf s
11	x rdfs:subClassOf y and y rdfs:subClassof z	x rdfs:subClassOf z
12	p rdf:type rdfs:ContainerMembershipProperty	p rdfs:subPropertyOf rdfs:member
13	o rdf:type rdfs:Datatype	o rdfs:subClassOf rdfs:Literal

reasoning process does. Given the above two triples, triple Ronaldo rdf:type Person can be easily derived. This is because Footballer is a subclass of Person, and every Footballer (including Ronaldo) is also a Person. Table 1 shows all the RDFS reasoning rules [12].

1.2 MapReduce Framework

MapReduce is a distributed programming model which is influenced by map and reduce functions from functional programming. Input/output of the map and reduce function is based on key-value pairs. Map function gets its input as key-value pairs and outputs intermediate key-value pairs. The values with the same key goes to the same reduce function which processes the results of these intermediate values [13]. A program may be deployed as many map and reduce functions distributed over cluster computers.

2 Hadoop RDFS Reasoning

The work by Urbani et al. [12] defines an execution order of the rules given in Table 1. Basically it starts with the execution of subproperty rules, then domain/range rules and finally subclass rules are executed. We follow this order (as summarized in Fig. 1) since it has been shown to be complete in the same work. Since duplicate triples can

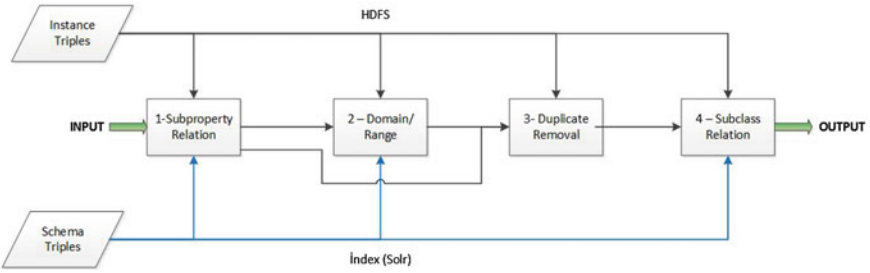


Fig. 1 Execution order of rules

be generated by subproperty and domain/range rules, duplicate removal is also an important step. Each of rule execution is handled by separate but connected in a pipeline map and reduce jobs. Output triples include all the input triples plus all the derived triples. Input and output triples are stored in Hadoop HDFS file system.

2.1 RDFS Reasoning with Triple Indexing

Since input and intermediate storage requirements are higher than that can fit in main memory, fast access to triples is needed during processing. For efficiency purposes, we propose to use indexing on schema triples, for which we utilize Apache Solr [14, 15]. The index server is configured so that triple subject, predicate and object are added to schema configuration.

Algorithm 1 shows the map and reduce jobs for reasoning from subproperty rules. Algorithms for domain/range rules, duplicate removal and subclass rules are not given due to space limitations. In all of the algorithms we basically follow the approach of [12] but we add indexing on top of it.

2.2 RDFS Reasoning with Hive

Most of the activities in RDFS reasoning are involved with managing and querying large datasets. Since Apache Hive is designed to efficiently manage and query large datasets, we employed it for implementing the jobs. Basic approach requires to define data table, import data and ask queries using Hql language. In our case, we have two tables named rdfsInput and rdfsSchema to store instance and schema triples, respectively. With this separation except the rules 12 and 13, all of the other rules can be implemented as simple join operations. Rules 12 and 13 are easier to handle without any joins. A sample Hive query (for Rule 5) looks like as shown in the following listing.

```
Setup(context)
connectSolrServer()
subproperties=getSchmemaTriples("rdfs: subPropertyOf")

Map(key,value)
if subproperties.contains(value.predicate) then
    key = "1" + value.subject + "-" + value.object
    emit(key, value.predicate)
end
if (subproperties.contains(value.object) and value.predicate == "rdfs:subPropertyOf") then
    key = "2" + value.subject
    emit(key, value.object)
end

Reduce(key,iterator values)
switch key[0] do
    case 1
        foreach predicate in values do
            superproperties.add(subproperties.recursiveGet(value))
        end
        foreach superproperty in superproperties do
            emit(null, triple(key.subject, superproperty, key.object)
        end
    case 2
        foreach predicate in values do
            superproperties.add(subproperties.recursiveGet(value))
        end
        foreach superproperty in superproperties do
            emit(null, triple(key.subject, "rdfs:subPropertyOf", superproperty)
        end
    end
endsw
```

Algorithm 1: Subproperty Job

```
<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>','
,rd2.object)
FROM rdfsSema rd1
LEFT OUTER JOIN rdfsSema rd2 ON (rd1.object=rd2.subject)
WHERE rd1.predicate=
'<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>'
AND rd2.predicate=
'<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>'
```

2.3 RDFS Reasoning with Pig

An alternative to Apache Hive is Apache Pig as far as the large dataset analysis is concerned. It offers a high-level language for creating pipeline through which data analysis is performed. In our case, we use two variables representing instance and schema triples and perform RDFS reasoning over them. A sample Pig code (for Rule 5) looks like as shown in the following listing.

```
AS (subject:chararray, predicate:chararray, object:chararray);
rdfsSema2 = LOAD '/user/rdfsSema/*.*' using PigStorage(' ')
AS (subject:chararray, predicate:chararray, object:chararray);
rdfsSemaFilter = FILTER rdfsSema BY
predicate ==
'<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>';
rdfsSemaFilter2 = FILTER rdfsSema2
BY predicate==
'<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>';
joined =JOIN rdfsSemaFilter BY object,rdfsSemaFilter2 BY subject;
result = FOREACH joined GENERATE CONCAT(rdfsSemaFilter::subject,'
<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>')
,rdfsSemaFilter2::object;
STORE result INTO '/user/resultPig/subproperty';
```

3 Experimental Evaluation

In this section, we present an experimental evaluation to understand the performance of the four approaches studied.

The experiment environment is a virtual machine Hadoop Cluster having five nodes, where each node has a dual core processor with 8 GB RAM and 250 GB disk storage. In the virtual machine configuration, each disk is dedicated to a single node to prevent disk access bottleneck.

We experiment with Dbpedia and Freebase datasets. Dbpedia dataset has 85 million triples. Our download of Freebase dataset originally has 2 billion triples but we also use a 33 million triple subset of it in the experiments. All the Freebase dataset results in the plots refer to this smaller dataset. All the results presented include duplicate removals where appropriate.

3.1 Experimental Results

3.1.1 Performance of the Subclass Relation

Figure 2a presents the efficiency on the subclass relations on Dbpedia and smaller Freebase datasets. From the figure, proposed indexing method gives the best result.

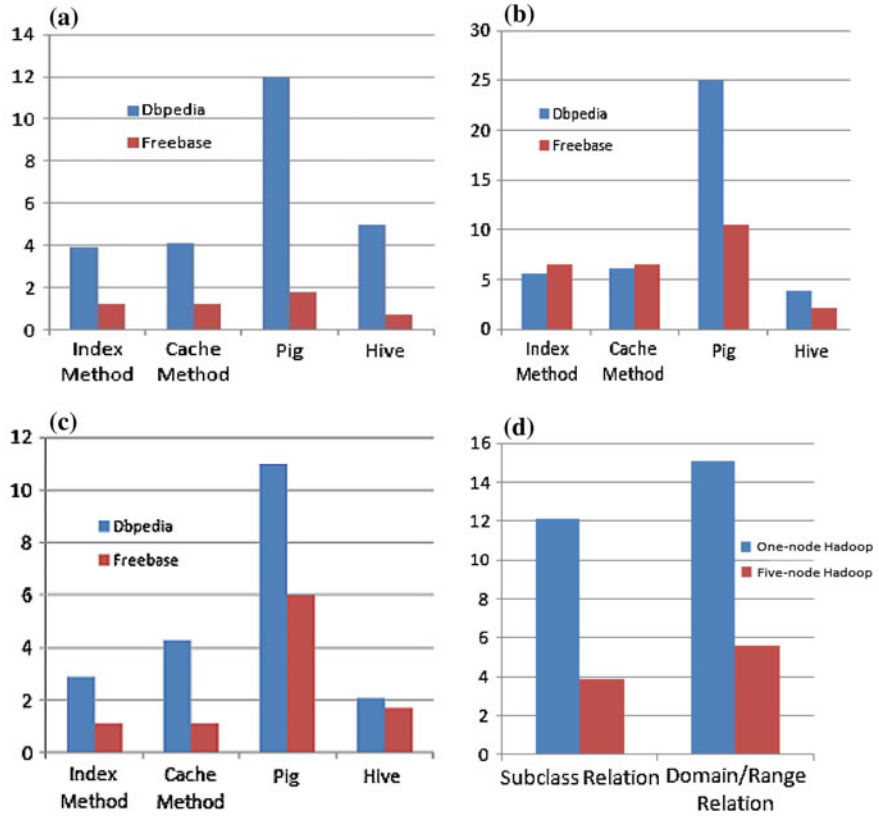


Fig. 2 Runtime (in minutes) of the methods on the two datasets: **a** subclass relations, **b** domain/range relations, **c** subproperty relations, **d** scalability

Hive and caching (indicated as Cache Method) attain good results too. Cache method is simply the caching of triples in the main memory. As a result of this reasoning 8.4 million new triples are derived from Dbpedia dataset. Since Hive and Pig do not have transitive closure queries, their result is not complete. Hence, they miss 35,000 out of 8.4 million triples with one iteration of joins. To overcome this problem, more join iterations are needed.

The utility of proposed indexing method becomes clearer as with complete Freebase dataset RDFS Reasoning is not possible with other methods due to high volume of data. With indexing the runtime takes 50 min on complete Freebase dataset. Indeed this is the main reason that we extract a subset of complete Freebase dataset in order to to have methods compared.

3.1.2 Performance of the Domain/Range Relation

The results for the domain/range kind of relations are shown in Fig. 2b. Where Hive gives the best result while Indexing method performs slightly better than Cache method. Pig is the slowest. As a result of domain/range relations 5 million new triples are derived from the Dbpedia dataset.

For the complete Freebase dataset, the reasoning process spends 10 h and creates 30 million new triples with Indexing method. Other methods can not process complete Freebase dataset.

3.1.3 Performance of the Subproperty Relation

Figure 2c shows the performance results on subproperty relations, where Indexing method and Hive give the best performance. As a result of the reasoning 2.3 million new triples are derived from Dbpedia dataset. Pig performs the worst on both datasets. The reason that Pig is the worst can be attributed to its general-purpose data pipeline design.

For the complete Freebase dataset the reasoning process spends 50 min with Indexing method. Again other methods can not process complete Freebase dataset.

3.1.4 Scalability

Figure 2d shows the scalability results for subclass and domain/range relations. The results indicate that for both kind of relations, the runtime performance improves with increased node counts. The runtime improves about 70 % with five node Hadoop cluster over single node Hadoop configuration. This shows the utility of MapReduce with Hadoop for RDFS reasoning.

4 Conclusion

In this work, we do RDFS Reasoning with MapReduce on Hadoop clusters. We proposed to use Indexing to improve the performance of RDFS reasoning and also developed algorithms to utilize other Hadoop technologies namely Apache Hive and Apache Pig for this task. Moreover, caching is an another method where new triples are derived in the memory and cached. Our indexing method uses Apache solr engine.

We experimented with two large datasets, Dbpedia and Freebase on a small Hadoop cluster. The results show that RDFS reasoning with large datasets on relatively small Hadoop cluster works in acceptable time. Moreover, the results show that Indexing method is highly scalable and also, on average, performs better than other methods experimented with. Our study also shows that as far as the Hadoop ecosystem technologies are concerned Hive is good choice for RDFS reasoning.

References

1. T. Berners-Lee, J. Hendler, O. Lassila, The semantic web. Sci. Am. **284**(5), 28–37 (2001)
2. J. Weaver, J.A. Hendler, Parallel materialization of the finite rdfs closure for hundreds of millions of triples. in: *Proceedings of the 8th International Semantic Web Conference* (ISWC 2009), pp. 682–697, Springer (2009)
3. M. Husain, L. Khan, M. Kantarcioglu, B. Thuraisingham, Data intensive query processing for large RDF graphs using cloud computing tools. in: *Proceedings of the IEEE 3rd International Conference on Cloud Computing* (CLOUD 2010), pp. 1–10, (2010)
4. Apache Hadoop, http://hadoop.apache.org/. Accessed April 2014
5. S.G.J. Dean,Mapreduce: simplified data processing on large clusters. in *6th Symposium on Operating Systems Design and Implementation* (OSDI 2004), (2004)
6. Apache Hive, http://hive.apache.org/. Accessed April 2014
7. Apache Pig, http://pig.apache.org/. Accessed April 2014
8. A. Thusoo, J.S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, R. Murthy, Hive-a petabyte scale data warehouse using Hadoop. in: *Proceedings of the IEEE 26th International Conference on Data Engineering* (ICDE 2010), pp. 996–1005, (2010)
9. P. Papailiou, I. Konstantinou, D. Tsoumakos, P. Karras, N. Koziris, H2RDF+: High-performance distributed joins over large-scale RDF graphs. in: *Proceedings of the IEEE International Conference on Big Data*, pp. 255–263, (2013)
10. S. Jianling, J. Qiang, Scalable RDF Store Based on HBase and MapReduce. in: *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering* (ICACTE 2010), pp. 633–636, (2010)
11. D. Brickley, R.V. Guha (eds.), RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, (2004)
12. J. Urbani, S. Kotoulas, E. Oren, F. Van Harmelen, Scalable distributed reasoning using mapreduce. in: *Proceedings of the 8th International Semantic Web Conference* (ISWC 2009), pp. 634–649, Springer (2009)
13. T. White, *Hadoop The Definitive Guide* (O'Reilly Media/Yahoo Press, Sebastopol, 2012)
14. Y. Zhanga, T. Chenb, W. Youc, J. Yud, J. Sune, H. Chenf, A new efficient semantic web platform based on the Solr, SIREn and RDF. in: *Proceedings of the International Conference on Information Engineering* (2012)
15. Apache Solr, http://lucene.apache.org/solr/. Accessed April 2014

Part VII
Performance Evaluation

On-Demand Prefetching Heuristic Policies: A Performance Evaluation

Olivia Morad and Alain Jean-Marie

Abstract Prefetching is a basic mechanism in the World Wide Web that speculates on the future behaviour of users to avoid the response delays. The relatively new requirement of the instantaneous response in some interactive services like On-Demand applications fuelled the need for ways to represent and reason about the challenging problem of prefetching control and performance evaluation. We study this challenging problem under a network protocol that adopts the simultaneous prefetching with equal-shared bandwidth, and in prefetching situations in which the controller seeks to reach a Zero-Cost system state as quickly as possible. Within this context, our first contribution is providing the backbone of a new paradigm for the performance evaluation of the On-demand prefetching policy. This backbone consists of our previously developed prefetching control model; the PREF-CT model and our previously developed optimal control algorithms; the ONE-PASS and the TREE-DEC algorithms. Our second contribution is developing the prefetching heuristic algorithm: the RBP. Compared to the optimal prefetching policies, the prefetching policies computed by our heuristic algorithm the RBP show significant performance in terms of the user's latency and the bandwidth utilization.

Keywords Optimal Control · Prefetching · Performance evaluation

1 Introduction

Prefetching is the speculative retrieval of a resource into a cache in the anticipation that it can be served from the cache in the future . [1]. Traditionally, the mechanisms developed so far in the literature to handle the prefetching problems oscillate between mechanisms that use some simple heuristic rules (e.g., Best-First . [2], etc) to

O. Morad (✉) · A. Jean-Marie
LIRMM : Montpellier 2 University/CNRS, Montpellier, France
e-mail: Olivia.morad@lirmm.fr

A. Jean-Marie
INRIA, 161 rue ADA, 34095 Montpellier, France
e-mail: Alain.Jean-Marie@inria.fr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_33

mechanisms that use more elaborated prediction technique to anticipate the next user requests (e.g., Markov models . [3, 4], etc) [5]. All these mechanisms are based on the execution digraph (or directed graph) where the nodes represent the tasks (e.g., web pages) and the arcs model the fact that a task can be executed once another has been done [6]. Indeed, the performance of the heuristic prefetching policies computed by most of these mechanisms has not been evaluated within the optimization context, and has not been compared to a known optimal prefetching policy (except in a first trial reported in [7]). Most of the reported prefetching optimizations are limited to a single step navigation, hence a short-sighted and an off-line optimization that results in a local optimal prefetching [6]. Modelling the prefetching problem as a control optimization problem in the Stochastic Dynamic Programming framework has not been well addressed in the literature; the first relevant attempt is reported in [7]. In the present work, we address the prefetching control problem in which the controller seeks to reach a *Zero-Cost state as quickly as possible (optimally)*. A *Zero–Cost state* prefetching control problem can be solved under different prefetching strategies; this depends on the costs imposed by the On-Demand system (e.g., degree of importance of the instantaneous response, latency, cache pollution, traffic congestion, etc…). The prefetching strategy determines how the controller will manage the utilization of a limited allocated bandwidth to achieve the minimization of some system's costs. We focus on the prefetching policies computed under the *prefetching acceleration strategy*. In this prefetching strategy, the controller maximizes the utilization of a limited allocated bandwidth while minimizing the user's blocking and latency costs incurred along the way. The*optimal prefetching acceleration policy*is the policy in which the controller achieves the *optimal balance* between the maximization of the bandwidth utilization and the minimization of the user's blocking and latency costs to reach a Zero-Cost state optimally. Indeed, the computation of this*optimal prefetching acceleration policy* that maximizes the bandwidth utilization in a manner that ensures the minimum user's blocking and latency costs can be solved through an optimization problem of a two-stage as follows:

1. Find prefetching policies that minimize the latency.
2. Among these prefetching policies, find those that maximize the bandwidth utilization.

The*optimal prefetching acceleration policy* computed through this two-stage optimization is out of the current scope and will be studied in our future research. In contrast, the current work focus on the paradigm of the performance evaluation that we proposed to evaluate the prefetching heuristic policies. In this paradigm, the evaluation is done through the comparison to the *optimal prefetching acceleration policy* computed based on the Negative Stochastic Dynamic program; in which the undiscounted total expected cost is minimized as a single global objective function.

The rest of the paper is organized as follows: Section 2 reviews briefly the main aspects of the optimal prefetching control policy for a *Zero-Cost state* control problem. Section 3 presents the Relevant Blocks Prefetching algorithm (RBP) that we propose. Section 4 reviews briefly the experimentation aspects. Finally, we conclude and highlight directions for future work.

2 PREF-CT: The Optimal Prefetching Control Policy

We model the Zero-Cost state control problem as a Negative Stochastic Dynamic Programming problem [8]. Therefore, we adopt the undiscounted expected total cost as our optimality criterion. Let $(s_0, s_1, ..., s_t, ...)$ denote the sequence of states, which is a random process. The cost we want to minimize is, for all state s,

$$J^{\pi}(s) = E_s^{\pi}\left\{\sum_{t=0}^{\infty} c(s_t, a_t)\right\}, \tag{1}$$

The control policy is π, and E_s^{π} the expectation of the random process under the application of this policy, conditioned on the fact that the state process starts from $s = s_0$.We identified two possible types of *Zero-Cost state(s)* in PREF-CT [5]: *the absorbing state* and *the full cache state*. In the first one, whenever the user accesses a specific block he remains viewing this block forever without any block changing. In the second, the cache is filled with all the blocks entirely and the user alternates the viewing between the blocks. Once one of these sets of states is entered the system stays in that set and no further costs are incurred thereafter. The assumptions of PREF-CT assure the existence of the optimal prefetching policy and assure that the Value Iteration algorithm converges to it in a finite number of iterations. We detected a significant special structure of acyclic Markov chain for the optimal prefetching policy. This special structure results in restricting the search on what we called: *useful policies* (all actions are *useful actions*). *A useful action* is an action that does not give control to block that already entirely exists in the cache, in addition this action should make progress i.e., an augmentation of the amount of information in the cache by actually prefetch at least one piece of block. This special structure allows deriving two optimality algorithms; the ONE-PASS [5] and the TREE_DEC [5], which improve the computations of the optimal prefetching policy. Both algorithms are based on the idea of that in acyclic graphs we can start from s^{dest} with $\upsilon\left(s^{dest}\right) = 0$,and perform a backward pass in which every state is visited after all its successor states have been visited. ONE-PASS computes the value function and the optimal policy in one scan of the state space; or in other words solving a system of equations in only one iteration. Whereas TREE_DEC computes the value function and the optimal policy based on a Markov tree decomposition in which sequential sets of systems of equations are solved (for more details the reader should refer to [4]).

3 The Heuristic Prefetching Control Policy

Despite the computations improvement achieved by the ONE-PASS and the TREE-DEC algorithms, the curse of dimensionality nevertheless exists. Therefore, an approximation of the *optimal useful* prefetching policy is necessary. The heuristic prefetching algorithm the RBP developed in this work tries to emulate the

controller's prefetching acceleration strategy efficiently based on the general aspects of this prefetching strategy.

The RBP prefetching heuristic algorithm:

The Relevant Blocks Prefetching algorithm, RBP, is based on the idea that at every system state $s = (b, p)$ there is a set of relevant variables which have the highest impact on the total cost and hence the optimal actions. In our prefetching algorithm RBP, the current block in view b is the principal relevant variable that specifies the remaining relevant variables, which are all the blocks b' belongs to the set of the directly reachable blocks B'_b. The main principle of the RBP prefetching algorithm is the categorization of the states and hence the categorization of the prefetching actions; indeed a categorized mapping of a state s to a decision rule $d(s)$ and then to an action a. The RBP prefetching algorithm checks the proportions' status p for the set of the relevant blocks B'_b in the cache and based on that, it categorizes the state s. In consequence it categorizes the control prefetching rule $d(s)$ and hence determines the prefetching action a. The RBP prefetching algorithm when applied to some state s, begins by specifying the different sets of blocks in this state$s$. The algorithm specifies six categories of sets of blocks:

- $\Re s$: The relevant blocks which are all the directly reachable blocks from the current one.
- $C\Re s$: The relevant blocks with complete proportions in the memory (entirely prefetched).
- $I\Re s$: The relevant blocks with incomplete proportions in the memory (partially prefetched).
- $N\Re s$: The non-relevant blocks which are all the indirectly reachable blocks from the current one.
- $CN\Re s$: The non-relevant blocks with complete proportions in the memory (entirely prefetched).
- $IN\Re s$: The non-relevant blocks with incomplete proportions in the memory (partially prefetched).

If the number of the incomplete blocks (relevant or non-relevant) is equal to or less than the available bandwidth R, the optimal prefetching action is to prefetch all of these blocks entirely in the memory. In contrast, if the number of the incomplete blocks is greater than the available bandwidth R, the algorithm tries to apply a mechanism to choose some of them for prefetching as follows:- The algorithm computes the sum of the current remaining proportions y of the incomplete relevant blocks $I\Re s$. Depending on this value y the algorithm determines which prefetching scenarios will be chosen. Two prefetching scenarios may occur depending on whether the available bandwidth R is enough to cover this amount y or not:

1. *First scenario*: If this amount y is equal or superior to the available bandwidth R, the algorithm prefetches all the incomplete relevant blocks $I\Re s$ and distributes the bandwidth equally between them.
2. *Second scenario*: In contrast if this amount y is less than the available bandwidth R, the algorithm tries to specify and prefetch blocks as much as possible to utilize

all the available bandwidth. The selection of additional blocks for prefetching – the set Ω– is done using a specific sub-algorithm; the Blocks Look Ahead –BLA– which selects the blocks depending on the state *s*under consideration. More than one sub-case can happen in this second scenario as follows:

- All the relevant blocks belong to the set $\Re s$ are entirely prefetched in the memory: If the size of the set $IN\Re s$ of the incomplete non-relevant blocks is less than or equal to the size of the set available bandwidth R, the algorithm prefetches entirely all of them. In contrast, if the size of the set $IN\Re s$ of the incomplete non-relevant blocks is greater than the available bandwidth, the algorithm tries to choose some of them. The algorithm computes the greatest remaining proportion among these incomplete non-relevant blocks $IN\Re s$, and divides the bandwidth R by this amount to specify how many blocks can be prefetched, indeed specifying the size of the set Ω (see (2)). The aim of the division in (2) is to ensure in the first priority the entire prefetching of the incomplete non-relevant block with the greatest remaining proportion in the memory. The following step is invoking the BLA sub-algorithm to determine the prefetched blocks, that is the set Ω.

$$|\Omega| = \frac{R}{Max_{i \in IN\Re_s}(1 - p_i)}. \tag{2}$$

- At least one of the relevant blocks is not in the memory: In this situation the algorithm computes the greatest remaining proportion among the incomplete relevant blocks $I\Re_s$, and divides the bandwidth R by this amount to specify how many blocks –|Ω|– could be added to the current specified set $I\Re_s$. The aim of the division in (3) is to ensure in the first priority the entire prefetching of all the incomplete relevant blocks $I\Re_s$.The following step is invoking the Block Look Ahead algorithm to determine the additional prefetched blocks –the set Ω– as we discuss below.

$$|\Omega| = \frac{R}{Max_{i \in I\Re_s}(1 - p_i)}. \tag{3}$$

The Block Look Ahead algorithm (BLA): This algorithm chooses the candidate blocks to be added –to the prefetching set *as*–for prefetching based on the following principles:

- Searches for the most probable blocks and gives control to blocks as long as they are not entirely prefetched in the cache.
- Chooses the most probable relevant block (entirely prefetched or partially prefetched) from the set of the current relevant blocks $\Re s$ of the current state s. For the chosen block, identifies its related set of the incomplete directly reachable blocks.
- Chooses as many blocks as possible –to be added to the prefetching set a_s– as long as the number of these chosen blocks does not exceed |Ω|.

- Chooses the next most probable relevant block from the set of the current relevant blocks $\Re s$ of the current state s. The algorithm repeats the process until searching over all the set of the relevant blocks $\Re s$, and tries reaching the target size $|\Omega|$.

If there is still enough room to add additional blocks –still not reaching the target size $|\Omega|$–, the BLA algorithm chooses among the incomplete non-relevant blocks $IN\Re s$. The choice is based on a calculation of the weights of the incomplete non-relevant blocks $IN\Re s$. The set $IN\Re s$ will be ranked in a decreasing order of the blocks' weights. For each incomplete non-relevant block i, we calculate its weight ω_i as the sum of the transition probabilities of the blocks that can reach it –rather than reachable by the block i. The greater the weight, the more important the block i, and hence the more probable is that this block i will be selected in the prefetching process. As in the previous principle, the BLA algorithm chooses from the set $IN\Re s$ as many blocks as possible and tries reaching the target size $|\Omega|$.

$$\omega_i = \sum_{j \in B} \beta_{ji}. \tag{4}$$

The RBP prefetching algorithm computes a dynamic deterministic aggressive prefetching policy π. The computed policy is dynamic since the RBP algorithm takes into account the dynamic unpredictable user behavior during a streaming video session. The computed policy is deterministic since it defines the same course of actions for the possible system states. The actions of the RBP algorithm are computed based on the prediction model of the user behavior $\beta_{bb'}$ as well as the network's bandwidth sharing protocol that governs the blocks' proportions in the cache memory. Finally, the computed policy is aggressive in terms of: first the amount of information prefetches (several blocks); second the "searching in depth" principle that early prefetches components reachable much later in the user navigation. The principle of *the computation technique* is the maximization of the bandwidth utilization. Thus, the computation technique is dedicated to specifying how many blocks can be prefetched, and how much bandwidth should be utilized to each block to minimize the wasted bandwidth units. *The computation technique* accomplishes this specification depending on the available bandwidth R, the proportions of the blocks in the cache p, and the network protocol. *The prediction technique* tries as much as possible to well respect the prediction model of the user profile as follows:

BF (best first) [2]**:** As the best first heuristic, the RBP chooses the directly reachable blocks –the relevant blocks $\Re s$– at the beginning of the algorithm. In contrast to the naïve BF heuristic, these most likely blocks will not be prefetched unless first examining their proportions' status in the cache.

Proportional heuristic [2]**:** The proportional heuristic chooses k blocks, or no more than k blocks, from among the directly reachable blocks. Whereas the RBP doesn't fix a priori the number of predicted blocks. The number of predicted blocks is specified by *the computation technique* just explained above, and it differs from state to state.

Sequential heuristic (Branch – Mainline) [9]**:** The RBP algorithm uses both the breadth first search – as the Branch heuristic– and the depth first search –as the Mainline heuristic– to choose the predicted blocks. The BLA algorithm loops over the set of the relevant blocks $\Re s$ sorted in descending order of the blocks' transition probabilities. Thus, this implies a breadth first search limited to one step of the graph of the user's Markov chain. Each breadth first search (i.e., for each relevant block$i$) is followed by a depth first search of one step to determine and examine the set of the directly reachable blocks B_i' of this relevant block i specified just by the breadth first search.

The BLA sub-algorithm: The look-ahead feature in the RBP algorithm do a deeper search to choose some weighted blocks.

The intention of this deeper search over the graph of the user's Markov chain, is to respect the principle of the computation technique that maximize the utilization of the bandwidth based on accurate calculation of the blocks' proportions status in the cache. The running time complexity of the RBP is $\mathrm{O}\left(|B| \log_2 |B|\right)$.

Observe that the categorization principle of the RBP algorithm permits clustering every group of states that share common proportions properties, therefore mapping every cluster C_{bp} to a prefetching action a rather than mapping every state s to a prefetching action a. This clustering property is of importance to reduce significantly the memory necessary to store the prefetching policy to the client of the On-demand system.

4 Experimentation

We conducted two main experiments; in the first experiment the controller does not impose a prefetching acceleration strategy. While in the second experiment the controller imposes a *specific*prefetching acceleration in which the immediate user's latency has the same weight as the bandwidth utilization (i.e. the controller can sacrifice some user's latency in the early stages of the system in order to achieve instantaneous response for future stages). Under each experiment, we used different user's profiles and for each user profile, we conducted experiments with different state space size.

We computed the value function of each heuristic policy using the ONE-PASS or the TREE-DEC algorithms, and then the value function of each heuristic policy has been compared to the optimal one. For each heuristic policy, the measures used for the value function vector are: the average (ATC), the average error (AV.E), the maximum error (Max.E), the standard error (STD.E), the percent of success ($p.s$ %), and the percent of zero value $p.\upsilon\,(s) = 0$ % (within the success set).

5 Conclusion and Future Work

In this paper, we present a new paradigm for the performance evaluation of the On-demand prefetching policy based on the context of the control optimization. Our prefetching control model PREF-CT permits the researchers and the practitioners to build easily its variants, and compute the optimal prefetching policy to evaluate their heuristic prefetching policies. Our proposed heuristic algorithm the RBP showed that it provides an efficient prefetching acceleration policy in terms of the bandwidth utilization and the user's latency. Therefore, the RBP heuristic algorithm can serve significantly the practitioners interested in the prefetching acceleration strategy under a network protocol that adopts the simultaneous prefetching with equal-shared bandwidth.

As a future work, we will manipulate our paradigm to conduct a two-stage optimization for the computation of the optimal prefetching policy under the prefetching acceleration strategy, this permit for further accurate evaluation and optimality approximation.

Acknowledgments This research has been funded by the French National Research Agency (ANR): project VOODDO number ANR-07-RIAM-0012.

References

1. B.D. Davison, *The design and evaluation of web prefetching and caching techniques* (The State University of New Jersey, Rutgers, 2002)
2. N. J. Tuah, M. Kumar, S. Venkatesh, Investigation of a prefetch model for low bandwidth networks, in *Proceedings of the 1st ACM international workshop on Wireless mobile multimedia*, pp. 38–47 (1998)
3. X. Dongshan, S. Junyi, A new markov model for web access prediction. Comput. Sci. Eng. **4**, 34–39 (2002)
4. M.A. Awad, I. Khalil, Prediction of user's web-browsing behavior: Application of markov model. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Trans. **42**, 1131–1142 (2012)
5. O. Morad, A. Jean-Marie, *Prefetching control for on demand contents distribution: A Markov decision process model*, IEEE 22nd International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems, Paris, France, 2014, (in press)
6. F.V. Fomin, F. Giroire, A. Jean-Marie, D. Mazauric, N. Nisse, To satisfy impatient web surfers is hard. Theor. Comput. Sci. **526**, 1–17 (2014)
7. R. Grigoras, V. Charvillat, M. Douze, Optimizing hypervideo navigation using a Markov decision process approach, in *Proceedings of the Tenth ACM International Conference on Multimedia*, pp. 39–48 (2002)
8. M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (Wiley, Hoboken, 2009)
9. E. P. Markatos, C. E. Chronaki, A top-10 approach to prefetching on the web, in *Proceedings of INET*, pp. 276–290 (1998)

An Improved Upper Bound for the Length of Preset Distinguishing Sequences of Distinguished Merging Finite State Machines

Canan Güniçen, Kemal İnan, Uraz Cengiz Türker and Hüsnü Yenigün

Abstract In an earlier work, we have studied a special class of FiniteState Machines (FSMs) called Distinguished Merging FSMs (DMFSMs) and showed that one can construct a Preset Distinguishing Sequence (PDS) for a DMFSM with n states, p input symbols, and r output symbols in time $O(n^4 + pn^2)$ of length no longer than $O(n^3)$. In this work, we improve the upper bound for the length of a PDS to $(n-1)^2$, and present an algorithm to construct such a PDS for a DMFSM in time $O(n^4 + pn^2)$ or in time $O(rn^3 + pn^2)$.

Keywords Model-based testing · Finite-state machines · Preset distinguishing sequence

1 Introduction

Testing is an important part of a development process of numerous reactive systems in various areas such as sequential circuits, lexical analysis, software design, communication protocols, object-oriented systems, web services, and in many others [1, 3, 5, 6, 14]. However, testing is typically expensive, manual, and error-prone. This has led to much interest in automating parts of testing and one of the most encouraging approaches to automation is model-based testing (MBT). In MBT, the automation is based on a model M (see, for example, [2, 4, 9]). The model might be a specification of the required behavior of the system under test or it might represent some aspect of

C. Güniçen · K. İnan · U.C. Türker · H. Yenigün (✉)
Sabanci University, 34956 Orhanli, Tuzla, Istanbul, Turkey
e-mail: gunicen@sabanciuniv.edu

K. İnan
e-mail: inan@sabanciuniv.edu

U.C. Türker
e-mail: urazc@sabanciuniv.edu

H. Yenigün
e-mail: yenigun@sabanciuniv.edu

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_34

interest. Given a model M, there is the potential to automatically analyze M in order to generate test cases and check that observed output is acceptable. Recent evidence gathered in an industrial project involving hundreds of testers suggests that the use of MBT can lead to significant benefits [9].

Most approaches to MBT typically use state-based models, where the semantics of the model is described in terms of states and transitions between states. There has thus been much interest in testing from finite-state machines (FSMs) (see, for example, [5, 7, 8, 11, 17, 18]). While test tools might allow the user to use richer languages, the models can usually be mapped to FSMs for analysis.

There have been many attempts to automatically generate test sequences from FSM models of systems [17]. Among these testing methods, a particular set of methods (e.g. [13]) are based on sequences called *distinguishing sequences* (DSs). By using DSs one can identify the current unknown state of the underlying FSM. ADS can be a preset or adaptive. If the input sequence is known before the experiment then it is a *Preset Distinguishing Sequence (PDS)* and if the input symbol is decided after the response of the FSM to the previous input then it is an *Adaptive Distinguishing Sequence (ADS)*. Lee and Yannakakis have reported that checking the existence of a PDS is a PSPACE-complete problem, whereas checking the existence of an ADS can be decided in polynomial time [16].

Therefore, one may argue that using an ADS is always preferable to using a PDS [12, 13]. However, there may be cases where a PDS is preferable or the use of an ADS may be impossible. For example, when one considers the use of an ADS/PDS in isolation (not inside a test sequence) to identify the state of a black box implementation, a PDS may allow the use of a simpler/cheaper test structure since adaptivity is not required. Also in such a case, the use of an ADS may not be even possible if there are timing constraints that cannot be met by using adaptive testing. Therefore, we believe that the PDS generation problem is interesting from a practical point of view as well.

Güniçen et. al. recently introduced a class of FSMs called *Distinguishing Merging FSMs* (DMFSM) [10]. For the practical relevance of DMFSMs, they examined the ACM/SIGDA benchmark test suite consisting of FSM specifications ranging from simple to advanced circuits obtained from the industry. They observe that among the specifications in this benchmark set that possess a PDS, half of them are DMFSMs. This observation validates that the class of DMFSMs is a practically relevant set of FSMs. They also report that one can construct a PDS for a DMFSM with n states and p input symbols and r output symbols in time $O(n^4 + pn^2)$ of length no longer than $O(n^3)$. In this work, we improve these bounds and show it is possible to construct a PDS for a DMFSM in time $O(rn^3 + pn^2)$ or again in time $O(n^4 + pn^2)$ with length no longer than $(n-1)^2$.

The paper is structured as follows. We start in Sect. 2 by presenting formal definitions of related terminology. In Sect. 3, we prove the main complexity results. Finally, Sect. 4 draws conclusions and discusses possible future work.

2 Preliminaries

A deterministic finite automaton (DFA) is defined as a 4-tuple $A = (S, X, D, \delta)$ where S is a finite set of states, X is a finite set of input symbols, $D \subseteq S \times X$ is domain of specification, and $\delta : D \to S$ is a transition function. A deterministic finite-state machine (FSM), on the other hand, is defined as a 6-tuple $M = (S, X, Y, D, \delta, \lambda)$ where S is a finite set of states, X is a finite set of input symbols, Y is a finite set of output symbols, $D \subseteq S \times X$ is a domain of specification, $\delta : D \to S$ is a transition function, and $\lambda : D \to Y$ is an output function.

The following definition apply to both DFAs and FSMs. When $(s, x) \in D$, the (input) x is said to be *defined at (state)s*. When a DFA A (resp. an FSM M) is at one of its states s, if an input x that is defined at s is applied, A (resp. M) performs a transition to the state $\delta(s, x)$. For an FSM M, also the output $\lambda(s, x)$ is produced during this state transition. An input sequence $\bar{x} = x_1 x_2 \ldots x_k$ is said to be *defined at (state)* s if there exists a sequence of states $s_1 s_2 \ldots s_k s_{k+1}$ such that $s = s_1$ and for each $i = 1, 2, \ldots, k$ the input x_i is defined at s_i and $\delta(s_i, x_i) = s_{i+1}$. We use $\Omega(s) \subseteq X^\star$ to denote the set of input sequences that are defined at s. For a given set $S' \subseteq S$ of states, $\Omega(S')$ denotes the set $\cap_{s \in S'} \Omega(s)$. A DFA A (resp. an FSM M) is said to be *completely specified* if $D = S \times X$. Otherwise it is called *partially specified*. Partially specified DFAs/FSMs can be used as behavioral models of a system where the behavior of the system is not defined for a certain state and a certain input. In the context of our work, FSMs/DFAs are not used as language acceptors. Therefore, the notion of initial and final states are not required in their definition.

A DFA A (resp. an FSM M) can be depicted as a directed graph $G = (V, E)$ where $V = S$, and $(u, v) \in E$ if there exists an input x such that $\delta(u, x) = v$. The edges are labeled by the input symbols for a DFA and by the input/output symbols for an FSM. Figures 1 and 2 are examples of a DFA and an FSM, respectively.

For both DFA and FSM, the transition function is extended to sequences of inputs (including the empty sequence ε) as usual in the following way: For a state s, an input

Fig. 1 An example DFA A_0

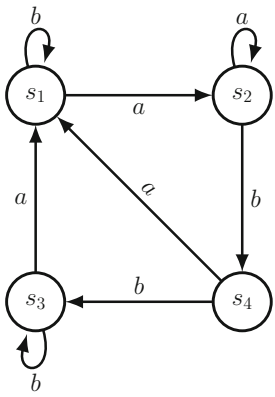
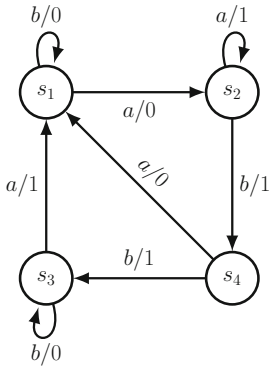


Fig. 2 An example FSM M_0



sequence $\bar{x} \in X^\star$, and an input symbol $x \in X$ such that $x\bar{x} \in \Omega(s)$, $\bar{\delta}(s, \varepsilon) = s$, $\bar{\delta}(s, x\bar{x}) = \bar{\delta}(\delta(s, x), \bar{x})$. In the case of an FSM, the output function is similarly extended and defined for the sequence $x\bar{x} \in \Omega(s)$ as $\bar{\lambda}(s, \varepsilon) = \varepsilon$ and $\bar{\lambda}(s, x\bar{x}) = \lambda(s, x)\bar{\lambda}(\delta(s, x), \bar{x})$. In remainder of the text (by abusing the notation), the symbols δ and λ are used to denote $\bar{\delta}$ and $\bar{\lambda}$, respectively. For a set S' of states, we also define $\delta(S', \bar{x}) = \{\delta(s, \bar{x}) \mid s \in S'\}$ and $\lambda(S', \bar{x}) = \{\lambda(s, \bar{x}) \mid s \in S'\}$ to extend the transition and output functions to a set of states.

We will now give some definitions that apply only to FSMs. Note that in this chapter, although we consider partially specified automata, only completely specified FSMs are considered. Therefore, the definitions below assume that $D = S \times X$, hence $\Omega(s) = X^\star$ for any state s.

Two states s and s' of an FSM M are said to be *equivalent* if for all $\bar{x} \in X^\star$, $\lambda(s, \bar{x}) = \lambda(s', \bar{x})$. Otherwise, when there exists an input sequence $\bar{x}$ such that $\lambda(s, \bar{x}) \neq \lambda(s', \bar{x})$, s and s' are said to be *distinguishable*. In this case, $\bar{x}$ is called *a separating sequence of s and s'*, or s and s' are said to be *distinguished (by $\bar{x}$)*. M is called *minimal* if for every two different states s and s', s and s' are distinguishable. Since it is always possible to find an equivalent minimal FSM M' in polynomial time, we only consider minimal FSMs in this chapter.

Definition 1 For an FSM $M = (S, X, Y, \delta, \lambda)$, a *Preset Distinguishing Sequence (PDS)* of M is an input sequence $\bar{D} \in X^\star$ such that for all states $s, s' \in S$, $s \neq s'$ implies $\lambda(s, \bar{D}) \neq \lambda(s', \bar{D})$.

Intuitively, a PDS $\bar{D}$ is an input sequence such that $\bar{D}$ is a separating sequence for any distinct pairs of states, which implies that every state produces a unique output sequence to $\bar{D}$. For FSM M_0 given in Fig. 2, aa is a PDS. It is clear that if M is not minimal, there does not exist a PDS for M. On the other hand, even when M is minimal, a PDS may not exist for M. Checking the existence of a PDS for an FSM is known to be PSPACE-complete [16, 19].

For two different states s and s' of an FSM, an input symbol x is said to be a *merging input for s and s'* if $\delta(s, x) = \delta(s', x)$. The input symbol x is said to be a *distinguishing input for s and s'* if $\lambda(s, x) \neq \lambda(s', x)$.

We now define an automaton A_M (with a sink state) that captures separability information for the pairs of states of an FSM M. The states of A_M (except the sink state) correspond to the unordered pairs $\{s, s'\}$ of states of M. The sequence of transitions from a state $\{s, s'\}$ of A_M to the sink state corresponds to a separating sequence for the states s and s'.

Definition 2 Let $M = (S, X, Y, D, \delta, \lambda)$ be an FSM. The *distinguishing automaton (DA)* A_M of M is an automaton $A_M = (S_A, X, D_A, \delta_A)$ which is constructed as follows:

- $S_A = \{\{s, s'\} \mid s, s' \in S \wedge s \neq s'\} \cup \{q^\star\}$. Intuitively, the set S_A consists of all 2-element subsets of S and an extra state $q^\star$.
- $D_A = \{(\{s, s'\}, x) \mid \lambda(s, x) \neq \lambda(s', x)$ or $\delta(s, x) \neq \delta(s', x)\} \cup \{(q^\star, x) | x \in X\}$. In other words, an input x is not defined at $\{s, s'\}$ if x is both merging and not distinguishing for s and s'. For $q^\star$, $(q^\star, x) \in D_A$ for all $x \in X$.
- For an input symbol x that is defined at a state $q \in S_A$, the transition function δ_A is defined as follows:

$$\delta_A(q, x) = \begin{cases} \{\delta(s, x), \delta(s', x)\} & \text{if } (q = \{s, s'\}) \wedge (\lambda(s, x) = \lambda(s', x)) \\ q^\star & \text{if } (q = \{s, s'\}) \wedge (\lambda(s, x) \neq \lambda(s', x)) \\ q^\star & \text{if } (q = q^\star) \end{cases}$$

As an example, the DA for FSM M_0 given in Fig. 2 is depicted in Fig. 3.

Even if M is completely specified, A_M can be partially specified. This is due to the fact that an input x is not defined at a state $\{s, s'\}$ if $\lambda(s, x) = \lambda(s', x)$ and $\delta(s, x) = \delta(s', x)$. Lemma 1 (the proof is available in Lemma 1 of [10]) explains how the DA A_M of M provides the separability information for the states of M.

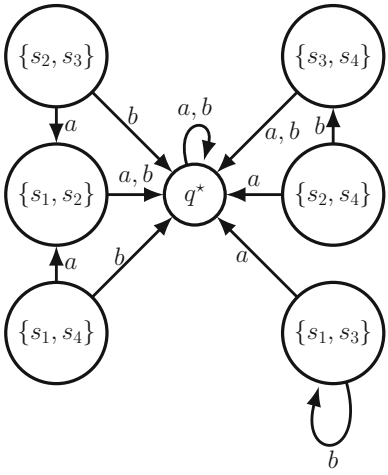


Fig. 3 The DA A_M of FSM M_0 in Fig. 2

Lemma 1 *Let $M = (S, X, Y, D, \delta, \lambda)$ be an FSM, $A_M = (S_A, X, D_A, \delta_A)$ be the DA of M, s_i and s_j be two different states of M. For an input sequence $\bar{x} \in X^\star$, $\lambda(s_i, \bar{x}) \neq \lambda(s_j, \bar{x})$ if and only if $\delta_A(\{s_i, s_j\}, \bar{x}) = q^\star$.*

We now introduce the special class of FSMs that will be studied in this work.

Definition 3 An FSM $M = (S, X, Y, D, \delta, \lambda)$ is said to be *distinguished merging (DM)* if for all states $s, s' \in S$ where $s \neq s'$, and for all input symbols $x \in X$, $\delta(s, x) = \delta(s', x)$ implies $\lambda(s, x) \neq \lambda(s', x)$.

Intuitively, an FSM M is DM if for any merging input x for states s and s', x is also a distinguishing input for s and s'. Hence, a merging input x for two different states s and s' is necessarily a distinguishing input for s and s'. We will call an FSM which is a DM as DMFSM. FSM M_0 given in Fig. 2 is a DMFSM.

In the rest of the paper, $M = (S, X, Y, D, \delta, \lambda)$ denotes a deterministic, completely specified and minimal DMFSM, $A_M = (S_A, X, D, \delta_A)$ denotes the DA of M. We will refer to the components of M and A_M (such as S, δ_A, etc.) without explicitly recalling their definitions. We also consistently use n, p, and r to refer to $|S|$, $|X|$, and $|Y|$, respectively.

3 Improved Upper Bound on the Length of PDSs

The following properties of DMFSMs have been studied in [10]: (i) A_M is completely specified, (ii) M always has a PDS, and (iii) a PDS of length no more than $(n^3 - 3n^2 + 4n - 2)/2$ can be computed in time $O(n^4 + pn^2)$. In this section, we present an algorithm to compute a PDS for M in time $O(n^4 + pn^2)$ or in time $O(rn^3 + pn^2)$. Furthermore, we show that the length of the PDS computed by the new algorithm cannot be longer than $(n - 1)^2$, which improves the upper bound provided by the algorithm of [10].

The algorithm to compute a PDS is given as Algorithm 1. However, we first introduce some notation that will be used and explain the approach of the algorithm. We use the term "block" to refer to a set of states. For an input sequence $\bar{x}$, the set of responses of the states in S to $\bar{x}$ is $\lambda(S, \bar{x})$. If two states s_i and s_j give the same response $\bar{y} \in \lambda(S, \bar{x})$, this means $\bar{x}$ cannot distinguish s_i and s_j. We use $S_{\bar{x}/\bar{y}}$ to denote the block that generates the output sequence $\bar{y}$ when the input sequence $\bar{x}$ is applied. In other words, $S_{\bar{x}/\bar{y}} = \{s \in S \mid \lambda(s, \bar{x}) = \bar{y}\}$.

Suppose that an input sequence $\bar{x}$ is applied to S, and two different states s_i and s_j produce the same response to $\bar{x}$. Since M is a DMFSM, we know that $\delta(s_i, \bar{x}) \neq \delta(s_j, \bar{x})$. Let $\bar{x}'$ be a separating sequence for $\delta(s_i, \bar{x})$ and $\delta(s_j, \bar{x})$, i.e., $\lambda(\delta(s_i, \bar{x}), \bar{x}') \neq \lambda(\delta(s_j, \bar{x}), \bar{x}')$. In this case, it is guaranteed that the combined sequence $\bar{x}\bar{x}'$ can separate s_i and s_j, i.e., we have $\lambda(s_i, \bar{x}\bar{x}') \neq \lambda(s_j, \bar{x}\bar{x}')$. This means that even though $\bar{x}$ cannot separate two states s_i and s_j, it is always possible to extend $\bar{x}$ to a separating sequence for s_i and s_j.

Considering that an input sequence $\bar{x}$ is applied to the states of M, the following definition characterizes the entire separation information of the states of M after the application of $\bar{x}$: $\pi(\bar{x}) = \{\delta(S_{\bar{x}/\bar{y}}, \bar{x}) \mid \bar{y} \in \lambda(S, \bar{x})\}$. Intuitively, for each possible output sequence $\bar{y} \in \lambda(S, \bar{x})$, there is a block B in $\pi(\bar{x})$, corresponding to the states (block) $S_{\bar{x}/\bar{y}}$. Instead of the block $S_{\bar{x}/\bar{y}}$, the block $\delta(S_{\bar{x}/\bar{y}}, \bar{x})$ is kept in $\pi(\bar{x})$, giving directly the set of states yet to be separated by extending $\bar{x}$. When $|\pi(\bar{x})| = n$, $\bar{x}$ is a PDS which is shown by Lemma 2.

Lemma 2 *For an input sequence $\bar{x}$, if $|\pi(\bar{x})| = n$ then $\bar{x}$ is a PDS for M.*

Proof The cardinality of $\pi(\bar{x})$ is equal to the cardinality of $\lambda(S, \bar{x})$ since there is one block in $\pi(\bar{x})$ for each $\bar{y} \in \lambda(S, \bar{x})$. Therefore $|\pi(\bar{x})| = n$ implies $|\lambda(S, \bar{x})| = n$, which means that there are n different responses from the states of M to $\bar{x}$. □

Note that the claim of Lemma 2 is valid for general FSM, even when M is not a DMFSM. We also have the following property for the blocks generated by an input sequence on the states of a DMFSM.

Lemma 3 *For an input sequence $\bar{x}$, $\sum_{B \in \pi(\bar{x})} |B| = n$.*

Proof For two different states s_i and s_j, if $\lambda(s_i, \bar{x}) \neq \lambda(s_j, \bar{x})$, then $\delta(s_i, \bar{x})$ and $\delta(s_j, \bar{x})$ are in different blocks of $\pi(\bar{x})$. If $\lambda(s_i, \bar{x}) = \lambda(s_j, \bar{x})$, then $\delta(s_i, \bar{x})$ and $\delta(s_j, \bar{x})$ are in the same block of $\pi(\bar{x})$ but in this case $\delta(s_i, \bar{x}) \neq \delta(s_j, \bar{x})$, since M is a DMFSM. Therefore, each state in each block of $\pi(\bar{x})$ corresponds to a unique state of M. □

Algorithm 1 computes a PDS $\bar{D}$ iteratively by extending an input sequence $\bar{x}$ progressively into an input sequence that can separate more and more states of S. The algorithm terminates when the condition given by Lemma 2 is met. In each iteration, the algorithm extends $\bar{x}$ by appending a separating sequence for at least one pair of yet to be separated states. Note that, for a block B in $\pi(\bar{x})$ where $|B| > 1$, there are at least two states s_i and s_j yet to be separated. The algorithm picks such a block B and two states s_i and s_j in B, and extends the current $\bar{x}$ by a separating sequence for s_i and s_j. In order to find a separating sequence for s_i and s_j, Algorithm 1 makes use of A_M. For a pair of states s_i and s_j, there exists a state $q = \{s_i, s_j\}$ in A_M. Based on Lemma 1, any input sequence $\bar{x}$ such that $\delta_A(q, \bar{x}) = q^\star$, $\bar{x}$ is a separating sequence for s_i and s_j. In order to keep the generated PDS short, Algorithm 1 prefers shortest input sequence $\bar{\tau}_{i,j}$ that can take the state $q = \{s_i, s_j\}$ in A_M to the state $q^\star$.

In the remaining of this section, we will show that (i) Algorithm 1 generates a PDS for M, (ii) it can be implemented to run in time $O(n^4 + pn^2)$ or $O(rn^3 + pn^2)$, and (iii) the length of the generated PDS is no longer than $(n-1)^2$.

Theorem 1 *Algorithm 1 computes a PDS for M.*

Proof Since M is deterministic, each state generates one output sequence as a response to an input sequence $\bar{x}$. We obviously have $|\pi(\bar{x})| = |\lambda(S, \bar{x})| \leq n$, for any input sequence $\bar{x}$. It is clear from line 5 of the algorithm that the algorithm

Algorithm 1: An algorithm to compute a PDS for a DMFSM

```
Input  : A DMFSM M = (S, X, D, δ, λ)
Output: A PDS D̄ for M
begin
1   compute the DA A_M of M;
2   for each state q = {s_i, s_j} of A_M, compute a shortest input sequence τ̄_{i,j} such that
    δ_A({s_i, s_j}, τ̄_{i,j}) = q*;
3   x̄ = ε;
4   π(ε) = {S};
5   while |π(x̄)| < n do
6       let B be a block in π(x̄) with |B| > 1;
7       let s_i and s_j be two different states in B;
8       compute π(x̄τ̄_{i,j}) using π(x̄);
9       x̄ = x̄ τ̄_{i,j};
10  return x̄ as D̄;
end
```

terminates when $|\pi(\bar{x})| \geq n$. Therefore, when the algorithm terminates we must have $|\pi(\bar{x})| = n$. Using Lemma 2, we can then state that when the algorithm terminates, it returns a PDS.

Note that initially $\pi(\bar{x}) = \{S\}$, hence $|\pi(\bar{x})| = 1$. We will now prove that the algorithm terminates eventually by showing that in each iteration $|\pi(\bar{x})|$ increases at least by one (hence eventually reaches to n).

Consider an iteration of the algorithm where the states s_i and s_j (belonging to a particular block B of $\pi(\bar{x})$) are picked at line 7. The states s_i and s_j correspond to two different states s_i' and s_j' such that $\delta(s_i', \bar{x}) = s_i$ and $\delta(s_j', \bar{x}) = s_j$. We also know that $\lambda(s_i', \bar{x}) = \lambda(s_j', \bar{x})$, since s_i and s_j belong to the same block B in $\pi(\bar{x})$. The algorithm uses $\bar{\tau}_{i,j}$ to extend $\bar{x}$ in this iteration. For $\bar{\tau}_{i,j}$ we know that $\lambda(s_i, \bar{\tau}_{i,j}) \neq \lambda(s_j, \bar{\tau}_{i,j})$. This also means $\lambda(s_i', \bar{x}\bar{\tau}_{i,j}) \neq \lambda(s_j', \bar{x}\bar{\tau}_{i,j})$. Hence the states $\delta(s_i', \bar{x}\bar{\tau}_{i,j})$ and $\delta(s_j', \bar{x}\bar{\tau}_{i,j})$ will be in two different blocks in $\pi(\bar{x}\bar{\tau}_{i,j})$, meaning that there will be more blocks in $\pi(\bar{x}\bar{\tau}_{i,j})$ then we have in $\pi(\bar{x})$. □

We will use the following lemma for the complexity analysis of Algorithm 1. The lemma suggests two different implementation approaches. The running time of the overall algorithm depends on the implementation approach used. A discussion on which approach needs to be used is given in Sect. 4.

Lemma 4 *For a given input sequence $\bar{x}$ and an input symbol x, $\pi(\bar{x}x)$ can be computed from $\pi(\bar{x})$ in time $O(n^2)$ or in time $O(rn)$.*

Proof We first show that for a block B in $\pi(\bar{x})$, the blocks in $\pi(\bar{x}x)$ that will be created from B can be computed in $O(|B|^2)$ or in time $O(r|B|)$. To obtain the time $O(|B|^2)$, one can consider the pairs of the states in B and compare their responses to the input symbol x, placing the states with the same response in the same block. To obtain the time $O(r|B|)$, one can first create an empty block B_y for each output

symbol $y \in Y$. A state $s \in B$ is placed into the block B_y iff $\lambda(s, x) = y$. After processing all the states in B in this way, an empty block B_y is removed.

We now show the claim of the lemma easily in the following way. Each block $B \in \pi(\bar{x})$ is handled in one of the ways explained above. If we process the blocks by pairwise comparison of the states, processing all the blocks in $\pi(\bar{x})$ takes time $\sum_{B\in\pi(\bar{x})} O(|B|^2) = O(n^2)$ since $\sum_{B\in\pi(\bar{x})} |B| = n$ by Lemma 3. If we process the blocks by creating empty blocks for each output symbol, then processing all the blocks takes time $\sum_{B\in\pi(\bar{x})} O(r|B|) = O(rn)$, again using Lemma 3.

Theorem 2 *Algorithm 1 can be implemented to run in time* $O(n^4 + pn^2)$ *or in time* $O(rn^3 + pn^2)$.

Proof Constructing A_M requires $O(pn^2)$ time, since for each pair of states $\{s_i, s_j\}$, and for each input symbol x, $\delta_A(\{s_i, s_j\}, x)$ needs to be computed. A_M has $O(n^2)$ states. Once A_M is constructed, a backward breadth-first search from $q^\star$ can be used to compute the shortest path $\bar{\tau}_{i,j}$ from each state $q = \{s_i, s_j\}$ of A_M to $q^\star$. Hence, step 2 can also be implemented to run in $O(pn^2)$ time.

As shown in the proof of Theorem 1, $|\pi(\bar{x})| = 1$ initially, and $|\pi(\bar{x})|$ increases by at least one in every iteration, and $|\pi(\bar{x})| = n$ when the algorithm terminates. Therefore, the while loop at line 5 can iterate at most $n - 1$ times.

During an iteration of the while loop, we need to pick a block B of size greater than 1. This can be accomplished in time $O(n)$ by going over the blocks of $\pi(\bar{x})$ which is known to have at most n blocks. After picking two states s_i and s_j from B, deciding $\bar{\tau}_{i,j}$ is just a lookup from the information recorded at line 2 of the algorithm. At this point, we want to point out that the length of the sequence $\bar{\tau}_{i,j}$ cannot be longer than $n - 1$, since for any two states s_i and s_j of a minimal FSM M with n states, there exists a separating sequence of length at most $n - 1$ (see e.g., [15]).

The most costly step of the proposed algorithm is line 8. If $\bar{\tau}_{i,j}$ is an input sequence of the form $x_1x_2 \ldots x_k$, one can first construct $\pi(\bar{x}x_1)$ from $\pi(\bar{x})$, then construct $\pi(\bar{x}x_1x_2)$ from $\pi(\bar{x}x_1)$, etc. Using this iterative approach, computing $\pi(\bar{x}\bar{\tau}_{i,j})$ from $\pi(\bar{x})$ can be performed in at most $n - 1$ iterations, since $|\bar{\tau}_{i,j}| \leq n - 1$. Using Lemma 4, this will take either $O(n^3)$ or $O(rn^2)$ time depending on the approach used for Lemma 4.

The overall cost of an iteration of the while loop is therefore either $O(n^3)$ or $O(rn^2)$, which makes the total cost of the while loop either $O(n^4)$ or $O(rn^3)$. Therefore, the total time for the algorithm is either $O(n^4 + pn^2)$ or $O(rn^3 + pn^2)$. □

Theorem 3 *Algorithm 1 computes a PDS of length at most* $(n - 1)^2$.

Proof Algorithm 1 iterates at most $n-1$ times, extending $\bar{x}$ by a separating sequence of length at most $n - 1$ in each iteration (see the proof of Theorem 2).

4 Concluding Remarks

For a DMFSM M with n states, p input symbols, and r output symbols, we presented an algorithm to compute a PDS of M, which is guaranteed to compute a PDS of length at most $(n-1)^2$, improving the current bound $O(n^3)$. We show that the algorithm given in this chapter can be implemented to run in time $O(n^4 + pn^2)$ or in time $O(rn^3 + pn^2)$. Depending on the relative values of n and r, one or the other implementation approach can be preferred. For a class of DMFSMs with $r = o(n)$, Algorithm 1 can be implemented in time $O(rn^3 + pn^2)$, which gives a better time performance than the algorithm given in [10].

As a future research direction, it would be interesting either to show that the bound $(n-1)^2$ is tight, or to find a lower and tight upper bound.

References

1. A.V. Aho, R. Sethi, J.D. Ullman, *Compilers: Principles, Techniques, and Tools* (Addison-Wesley, Reading, 1986)
2. M. Barnett, W. Grieskamp, L. Nachmanson, W. Schulte, N. Tillmann, M. Veanes, Towards a tool environment for model-based testing with AsmL. in *Formal Approaches to Testing, volume 2931 of Lecture Notes in Computer Science*, pp. 252–266, Montreal, Canada, 2003. Springer-Verlag
3. R.V. Binder, *Testing Object-Oriented Systems: Models, Patterns, and Tools* (Addison-Wesley, Reading, 1999)
4. E.G. Cartaxo, P.D.L. Machado, F.G.O. Neto, On the use of a similarity function for test case selection in the context of model-based testing. Softw. Test. Verif. Reliab. **21**(2), 75–100 (2011)
5. T.S. Chow, Testing software design modeled by finite-state machines. IEEE Trans. Softw. Eng. **4**(3), 178–187 (1978)
6. A.D. Friedman, P.R. Menon, *Fault Detection in Digital Circuits*, Computer Applications in Electrical Engineering Series (Prentice-Hall, Englewood Cliffs, 1971)
7. E. Gelenbe, Regular expressions and checking experiments. Technical Report AD0666696, Polytechnic Institute of Brooklyn NY Microwave Research Institue, September 1967. Defense Technical Information Center
8. G. Gönenç, A method for the design of fault detection experiments. IEEE Trans. Comput. **19**(6), 551–558 (1970)
9. W. Grieskamp, N. Kicillof, K. Stobie, V.A. Braberman, Model-based quality assurance of protocol documentation: tools and methodology. Softw. Test. Verif. Reliab. **21**(1), 55–71 (2011)
10. C. Gunicen, K. Inan, U.C. Turker, H. Yenigun, The relation between preset distinguishing sequences and synchronizing sequences. Formal Aspects of Computing (2014)
11. F.C. Hennie, Fault-detecting experiments for sequential circuits. in: *Proceedings of Fifth Annual Symposium on Switching Circuit Theory and Logical Design*, pp. 95–110, Princeton, New Jersey, November 1964
12. R.M. Hierons, G.V. Jourdan, H. Ural, H. Yenigun, Using adaptive distinguishing sequences in checking sequence constructions, in R.L. Wainwright and H. Haddad, editors, *SAC*, pp. 682–687. ACM, Mar 2008
13. R.M. Hierons, G.V. Jourdan, H. Ural, H. Yenigun, Checking sequence construction using adaptive and preset distinguishing sequences, in D.V. Hung and P. Krishnan, editors, *SEFM*, pp. 157–166. IEEE Computer Society, 2009
14. G.J. Holzmann, *Design and Validation of Computer Protocols* (Prentice Hall, Englewood Cliffs, 1991)

15. Z. Kohavi, *Switching and Finite State Automata Theory* (McGraw-Hill, NY, 1978)
16. D. Lee, M. Yannakakis, Testing finite-state machines: State identification and verification. IEEE Trans. Comput. **43**(3), 306–320 (1994)
17. D. Lee, M. Yannakakis, Principles and methods of testing finite-state machines—a survey. Proc. the IEEE **84**(8), 1089–1123 (1996)
18. A. Petrenko, N. Yevtushenko, Testing from partial deterministic FSM specifications. IEEE Trans. Comput. **54**(9), 1154–1165 (2005)
19. I.K. Rystsov, Polynomial complete problems in automata theory. Inf. Process. Lett. **16**(3), 147–151 (1983)

Time Parallel Simulation for Dynamic Fault Trees

T.H. Dao Thi, J.M. Fourneau, N. Pekergin and F. Quessette

Abstract Dynamic Fault Trees (DFT) are a generalization of Fault Trees which allow the evaluation of the reliability of complex and redundant systems. We propose to analyze DFT by a new version of time-parallel simulation method we have recently introduced. This method takes into account the monotonicity of the sample-paths to derive upper and lower bounds of the paths which become tighter when we increase the simulation time. As some gates of the DFT are not monotone, we adapt our method.

1 Introduction

Fault Tree analysis is a standard technique used in reliability modeling. Dynamic Fault Trees (DFT) are an extension of Fault Trees to model more complex systems where the duration and the sequences of transitions are taken into account. For a presentation of DFTs, one can refer to the NASA presentation [6]. DFTs are much more difficult to solve than static Fault Trees. Thus, new resolution methods have to be proposed. Fault Trees are composed of a set of leaves which model the components of the systems and some gates whose inputs are connected to the leaves or to the outputs of other gates. The value of the leaves is a boolean which is True if the component is down. The whole topology of the connection must be a tree. The root of the tree is a boolean value which must be True when the system has failed. The fault trees contain three types of gates: OR, AND, and K out of N (or voting) gates. All of them are logical gates we do not present here. DFTs allow four new types

J.M. Fourneau (✉) · F. Quessette · T.H. Dao Thi
PRiSM, CNRS UMR 8144, Versailles, France
e-mail: jmf@prism.uvsq.fr

T.H. Dao Thi
VIASM, Vietnam Institute for Advanced Study in Mathematics, Hanoi, Vietnam

N. Pekergin
LACL, Université de Paris Est Créteil, Créteil, France

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_35

of gate: priority AND (PAND), functional dependency (FDEP), sequential failures (SEQ), and SPARE gates. We first present the 4 gates added in the DFT framework and we introduce a Markov model of such a system. We assume that the failure times and the repair times follow exponential distributions. The four gates are:

- SPARE gate. It is used to represent the replacement of a primary component by a spare with the same functionality. Spare components may fail even if they are dormant but the failure rate of a dormant (λ^d) is lower than the failure rate of the component in operation (λ^a). A spare component may be "cold" if its failure rate is 0 while it is dormant, "hot" if the dormant has the same failure rate as an operating one, and it is called "warm" otherwise.
- FDEP. The FDEP gate has one main input connected to a component or another gate and it has several links connected to components. When the main input becomes True, all the components connected by the links must become True, irrespective of their current value.
- PAND. The output of the PAND gate becomes True when all of its inputs have failed in a preassigned order (from left to right in graphical notation). When the sequence of failures is not respected, the output of the gate is False.
- SEQ. The output of the SEQ gate becomes True when all of its inputs have failed in a preassigned order but it is not possible that the failure events occur in another order.

We assume that all the rates are distinct, therefore it is not trivial to lump the Markov chain of the DFT. In some sense, we are interested to solve the hardest model of the Markov chain associated to the DFT. We also assume that the graph of the connection when we remove the FDEP gates is a tree: no leaves are shared between two subtrees. The DFT is represented by a function F (the so-called structure function [5]) and vector $(X_1, \ldots, X_n, W_1, \ldots, W_p)$ where n is the number of components of the models (and leaves of the DFT) and p is the number of PAND gates in the model. X_i represents the state of component i. It is equal to False (resp. True) when the component is operational (resp. failed). W_k is associated to PAND gate with index k. It is True if the first component fails before the second one. Function F applied to state $(X_1, \ldots, X_n, W_1, \ldots, W_p)$ returns True when the system is down and False when it is operational. It is the value carried by the root of the DFT.

Due to these new gates, the static analysis based on cut sets and the Markov chain approach are much more difficult to apply. New techniques have been proposed (Monte Carlo simulation [8], process algebra [1]) but there is still a need for some efficient methods of resolution for large and complex DFT. We advocate that we can take into account the parallelism of our multicore machines and the monotone properties of many DFT models to speed up the simulation and obtain quantitative results in an efficient manner.

We make the following assumptions. The repairing rates do not depend on the state of the system. It is equal to μ_i for component i. When the input of an FDEP gate is repaired, it does not have any effect on the other components connected to the gate. These elements which have failed due to an event propagated by a FDEP gate are repaired independently after a race condition. Similarly, the components connected

to a SEQ gate fail in a specified order but they are repaired in a random order due to the race between independent repairing event. The chapter is organized as follows. In Sect. 2, we present the time-parallel simulation approach and the method we have proposed to speed up this technique when the system is monotone. In Sect. 3, we show how we can adapt the methodology to DFTs.

2 Time-Parallel Simulation

We now briefly present Nicol's approach for time-parallel simulation with iteration to fix the inconsistency of the paths built in parallel [7] and our extension to speed up the simulation of monotone systems [3, 4]. Let K be the number of logical processes (LP). The time interval $[0, T)$ is divided into K equal intervals $[t_i, t_{i+1})$. Let $X(t)$ be the state at time t obtained through a sequential simulation. The aim is to build $X(t)$ for t in $[0, T)$ through an iterative distributed algorithm. For the sake of simplicity, we assume that for all i between 1 and K, logical process LP_i simulates the ith time interval. The initial state of the simulation is known and is used to initialize LP_1. During the first run, the other initial states are chosen at random or with some heuristics. Simulations of the time intervals are ran in parallel. The ending states of each simulation are computed at the end of the simulation of the time intervals and they are compared to the initial state we have previously used. These points must be equal for the path to be consistent. If they are not, one must run a new set of parallel simulations for the inconsistent parts using the new point as a starting point of the next run on logical process LP_{i+1}. These new runs are performed with the same sequence of random inputs until all the parts are consistent.

Performing the simulation with the same input sequence may speed up the simulation due to coupling. Suppose that we have stored the previous sample-paths computed by LP_i. Suppose now that for some t, we find that the new point $a(t)$ is equal to a formerly computed point $b(t)$. As the input sequence is the same for both runs, both sample-paths have now merged: Thus, it is not necessary to build the new sample-path. Such a phenomenon is defined as the coupling of sample-paths. Note that it is not proved that the sample-paths couple and this is not necessary for the proof of the TPS that it happens. Indeed round i, it is proved by induction on i that LP_i is consistent. Clearly, coupling allows to speed up the computations performed by some LP and also reduces the number of rounds before global consistency of the simulation. For instance, in the left part of Fig. 1, the exact sample-path is computed after one round of fixing due to some couplings.

3 Improved Time-Parallel Simulation of Monotone DFT

We have shown in [3] how to use the monotone property exhibited by some models to improve the time-parallel approach. First, we perform an uniformization of the simulation process to obtain a discrete-time model because the approach is based

on the Poisson calculus methodology [2]. We consider a Poisson process with rate δ which is an upper bound of the transition rate out of any state: $\delta = \sum_{i=1}^{n}(\mu_i + \lambda_i)$, where μ_i is the reparation rate of component i and λ_i is the maximum of the failure rates of component i. Most of the component has a unique failure rate but a component connected to a warm or cold SPARE has two failures rates: one when is it dormant and one when it is in operation. Note that these rates may also be 0 when we model a cold SPARE or when the component is not repairable. The time instants t_n are given by this Poisson process and a random number u_n is used to draw the event which is realized at time t_n. Now we have to define the ordering.

Definition 1 (*Ordering*) We assume that False $<$ True and we define the following ordering on the states:
$(X_1^a, \ldots, X_n^a, W_1^a, \ldots, W_p^a) \leq (X_1^b, \ldots, X_n^b, W_1^b, \ldots, W_p^b)$ if for all i, $X_i^a \leq X_i^b$ and for all j, $W_j^a \leq W_j^b$. Note that it is not a total order.

Now we use an event representation of the model. Events are associated with transitions. The basic events in the DFT is the failure and the reparation of any component. Let e be an event. $P_e(x)$ is the probability that event e occurs at state x and $e(x)$ is the state reached from state x when event e occurs. It is more convenient that some events do not have any effect (for instance, the failure event will be a loop when it is applied on an already failed component).

Definition 2 (*Event monotone*) the model is event monotone if for all event e, $P_e(x)$ does not depend on state x and for all event e, if $\mathbf{x_1} \leq \mathbf{x_2}$ then $e(\mathbf{x_1}) \leq e(\mathbf{x_2})$.

We assume that the model is event monotone and that there exist two states *Min* and *Max* which are, respectively, the smallest and the largest of all states. We perform the time-parallel simulation as follows. We proceed with an initial run and with some runs for fixing the paths using the same sequence of random variables as in the first run. During the first run, we build two simulations on each processor (except the first one), one initialized with *Min* and another one with *Max*. The first process receives as usual the true initialization of the simulation process. As the model is event monotone, if both sample-paths couple (as within LP3 in the right part of Fig. 1), we know that

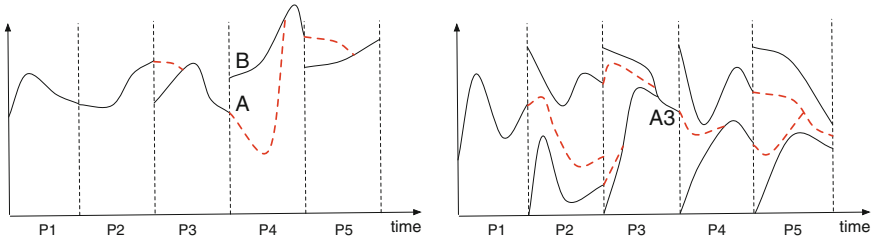


Fig. 1 *Left* TPS, coupling and fixing the sample-path. *Right* TPS of monotone systems with two bounding sample-paths and coupling. In both cases, the simulation is performed on five processors and the initial paths are in *black* while the correction step is in *dotted red lines*

the following of the paths does not depend on the initial state. When the paths do not couple, we obtain new upper and lower bounds for the next run (for instance in the right part of Fig. 1 the second run on LP3 uses the new bounds obtained by LP2 at the first run).

The improved version has three main advantages: first, at each iteration we have upper and lower bounds of the exact path, second the coupling of some paths give some correct information on the future and the time for correction decreases, and third at each iteration of the correction process the bounds become more accurate, see [4] for more details.

As noticed in [9], PAND gates are more complex to deal with than the other parts of a DFT. Let us first consider a PAND gate of two inputs, A and B. It is represented by vector (X_A, X_B, W_1).

Property 1 *The PAND gates is not event monotone.*

Proof Consider states (F, F, F) and (F, T, F). We clearly have $(F, F, F) \leq (F, T, F)$. Assume that event "failure of component A" occurs. The states become, respectively, (T, F, T), and (T, T, F). But $(T, F, T) \leq (T, T, F)$ does not hold. The model of a PAND gate is not event monotone.

To consider DFT with some PAND gates, we will need a more complex method that we will briefly introduce in the conclusions. Until then, we only consider DFT without PAND gates.

Property 2 *The structure function of a Dynamic Fault Tree which does not contain PAND gates is non decreasing.*

Proof Due to the tree topology, it is sufficient to prove that the output of an arbitrary gate or the links connected to a FDEP gate are not decreasing with the inputs of the gates. The structure function associated with static trees are nondecreasing with the ordering we consider. Thus, we just have to consider the three new gates: SEQ, SPARE and FDEP. For the structure function, SPARE and SEQ gates are similar to an AND gate. They only differ by the rates of the transition which are state dependent. The FDEP gate is a synchronization failure of several components. It changes the state $\mathbf{x}$ but it does not appear in function F. The structure function is not decreasing.

Definition 3 Let $\mathbf{x} = (X_1, ..., X_n)$ be an arbitrary state, we denote by $\mathbf{x} \odot 1_i$ the state $\mathbf{y} = (Y_1, ...Y_n)$ such that $Y_j = X_j$ for all $j \neq i$ and $Y_i = True$. Similarly, $\mathbf{x} \odot 0_i$ is defined by $Y_j = X_j$ for all $j \neq i$ and $Y_i = False$.

Property 3 *The model of a Dynamic Fault Tree which does not contain PAND gates is event monotone.*

Proof We have two families of event: failure and repair. We must check two conditions: the probability must be constant and the state reached after the occurrence of an event must be comparable if there were comparable before.

- Repairing of component i: The rate is μ_i. Thus the probability of repairing component i is μ_i/δ, which does not depend on the state.
  Now consider two states $\mathbf{x}$ and $\mathbf{z}$ such that $\mathbf{x} \leq \mathbf{z}$. Thus, $X_j \leq Z_j$ for all j. We reached states $\mathbf{u}$ and $\mathbf{v}$ from states $\mathbf{x}$ and $\mathbf{z}$ after the occurrence of event. Clearly, we get: $U_j = X_j, \forall j \neq i$ and $U_i = False$ and $V_j = Y_j, \forall j \neq i$, and $V_i = False$. Therefore, $\mathbf{u} \leq \mathbf{v}$. The event is monotone.
- Failure: We have a problem with components acting as a cold or warm spare. Indeed they do not have the same failure rates when they are dormant or inactive. Thus, we decompose the event in the following way. We consider a SPARE gate with only two components: a primary component A and a spare B. We decompose the event "failure of B" into two events $f1$ and $f2$. $f1(x)$ is the failure when component i is dormant: $Pr(event \quad f1) = \frac{\lambda_i^d}{\delta}$, and $f1(\mathbf{x}) = \mathbf{x} \odot 0_i$. $f2(x)$ is the extra failure event. It only occurs when component i is active, $Pr(event \quad f2) = \frac{(\lambda_i^a - \lambda_i^d)}{\delta}$, and $f2(\mathbf{x}) = \mathbf{x}$ if x_j is False and $f2(\mathbf{x}) = \mathbf{x} \odot 0_i$ if x_j .
  Clearly event $f1$ is monotone. Indeed, if $\mathbf{x} \leq \mathbf{z}$ then $\mathbf{x} \odot 0_i \leq \mathbf{z} \odot 0_i$. Now consider event $f2$. Assume that $\mathbf{x} \leq \mathbf{z}$. If the primary component associated to spare i is down at state $\mathbf{x}$, it also holds for state $\mathbf{z}$ because $X_i \leq Z_i$. Therefore, event $f2$ has the same effect on states $\mathbf{x}$ and $\mathbf{z}$. As $\mathbf{x} \odot 0_i \leq \mathbf{z} \odot 0_i$, the result holds in that case. If the primary component is up at state $\mathbf{x}$, we have $f2(\mathbf{x}) = \mathbf{x}$. And $\mathbf{z} \leq f2(\mathbf{z})$ as $f2$ is a failure event. Finally, $f2(\mathbf{x}) = \mathbf{x} \leq \mathbf{z} \leq f2(\mathbf{z})$, and we get: $f2(\mathbf{x}) \leq f2(\mathbf{z})$, the result holds as well.

Due to the former properties, one can perform an improved time-parallel simulation for DFT without PAND gates using the approach published in [4].

4 Some Results and Some Improvements

We give preliminary results on the coupling time of the simulation. We have simulated a large DFT with 1,000 leaves. We restrict ourselves to DFT without PAND gates as mentioned earlier in the chapter. We have assumed that the average repair time is always 10 times smaller than the time between failures. The failure rates are all equal to simplify the generation of the test. We present in Fig. 2 the sample-paths of the

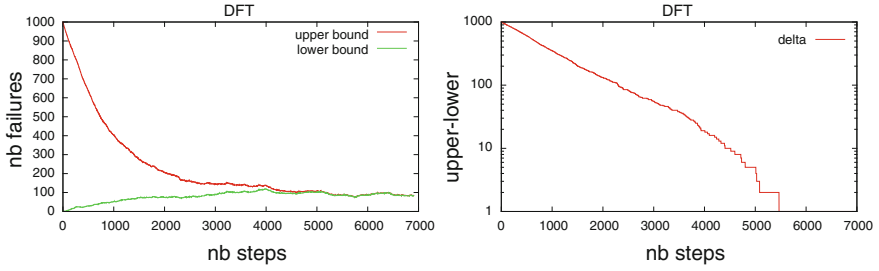


Fig. 2 *Left* upper and lower paths for a DFT with 1,000 leaves without PAND and FDEP gates. *Right* Difference in the number of failed components in upper and lower paths, in logarithmic scale

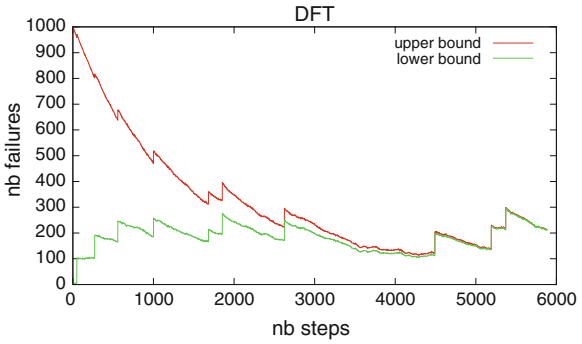


Fig. 3 Upper and lower paths for a DFT with 1,000 leaves and 20 FDEP gates connected to 100 random leaves

upper and lower bounds. We observe that the coupling of the paths occur after 6,900 time units. Here the coupling is defined by the property that all the components of the DFT are in the same state for the upper and the lower paths.

We have ran 500 independent experiments on random DFT with the same size to obtain an estimate of the expectation of the time before coupling which is equal to 7,400 time units. We represent in the right part of Fig. 2, in logarithmic scale, the difference in the number of failed components between the upper and the lower paths. Clearly, the figure shows a geometric convergence.

We now show the impact of the FDEP gates. We still consider DFT with 1,000 leaves. The FDEP gates are randomly assigned on the leaves. We have assumed very large output gates for the FDEP to emphasize the effect. As it can be seen in Fig. 3, the sample-path is less regular and the occurrence of an FDEP has a visible impact on the path. Furthermore, the mean time before coupling is smaller.

We now extend our approach to consider PAND gates. We use the same technique already known for static Fault Trees with repeated events. When we repeat an event, the topology is not a tree anymore as there exist two paths from the leaf to the root. Typically, when the number of such leaves is small, one can solve the model after conditioning on the states of the leaves. But for numerical computations we have to consider 2^m subtrees if there are m multiconnected leaves in the FT. We use the same idea, the PAND gates are simulated first and their results are inserted in the simulation as they are replaced by a virtual component. We decompose the DFT into subtrees. Each subtree is rooted by a PAND gate. We compute S the sum of the probability of the events occurring in the subtrees. Now, we change the initial step of the simulation. When we draw the random number, we check if $u_n < S$. In that case we trigger the event in the subtree and we compute the value of the PAND gate. If $u_n > S$, we just write u_n into the sequence. Then, we can begin a simulation with the PAND gate replaced by a component whose instants of failure and repairing have already been computed. This second part of the simulation may be performed in a time-parallel manner as formerly presented.

Acknowledgments This work was partially supported by grant ANR MARMOTE (ANR-12-MONU-0019).

References

1. H. Boudali, P. Crouzen, M. Stoelinga, Dynamic fault tree analysis using input/output interactive markov chains, in *The 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2007*, Edinburgh, UK, pp. 708–717 (2007)
2. P. Brémaud, *Markov Chains: Gibbs fields, Monte Carlo Simulation and Queues* (Springer-Verlag, New York, 1999)
3. J.-M. Fourneau, I. Kadi, N. Pekergin, Improving time parallel simulation for monotone systems, in *13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*, ed. by S.J. Turner, D. Roberts, W. Cai, A. El-Saddik, Singapore, pp. 231–234 (2009)
4. J.-M. Fourneau, F. Quessette, Tradeoff between accuracy and efficiency in the time-parallel simulation of monotone systems, in *EPEW 2012*, Munich (2012)
5. G. Merle, J.-M. Roussel, J.-J. Lesage, Algebraic determination of the structure function of dynamic fault trees. Reliab. Eng. Syst. Saf. **96**(2), 267–277 (2011)
6. NASA. Fault tree handbook, nureg-0492, technical report, United States Nuclear Regulatory Commission (1981)
7. D. Nicol, A. Greenberg, B. Lubachevsky, Massively parallel algorithms for trace-driven cache simulations. IEEE Trans. Parallel Distrib. Syst. **5**(8), 849–859 (1994)
8. K.D. Rao, V. Gopika, V.V.S.S. Rao, H.S. Kushwaha, A.K. Verma, A. Srividya, Dynamic fault tree analysis using monte carlo simulation in probabilistic safety assessment. Reliab. Eng. Syst. Saf. **94**(4), 872–883 (2009)
9. T. Yuge, S. Yanagi, Quantitative analysis of a fault tree with priority and gates. Reliab. Eng. Syst. Saf. **93**(11), 1577–1583 (2008)

On Performance Evaluation of Loss-Based Overload Control Mechanism in Signaling System with SIP Protocol

Halina Tarasiuk and Jan Rogowski

Abstract In this paper, we compare two approaches of deploying a loss-based overload control mechanism in a signaling system with the SIP protocol, hop-by-hop, and end-to-end. For deploying these mechanisms, we propose a simple, non-preemptive priority queueing scheme for a SIP server. In this scheme, overload control messages receive high priority while other signaling messages are served with a low priority. This approach allows effective support of performance of the loss-based overload control mechanism in the system. To get knowledge about advantages of the proposed solution, we compare both approaches (hop-by-hop and end-to-end) with a system without overload control. The measured performance metrics are related to average call setup time and goodput.

Keywords SIP protocol · Signaling system · Overload control mechanisms

1 Introduction

Nowadays, we can observe many efforts to deploy the Next Generation Networks (NGN) architecture in operational telecommunication networks [3]. ITU-T recommendations of the architecture and IETF specifications of signaling protocols allow us to implement a comprehensive signaling system to support quality of service in the network [8]. In addition, algorithms and methods such as admission control, resource allocation and provisioning, developed for voice over IP or IPTV services are available. However, based on our best knowledge, the still-missing element is how to deploy overload control mechanisms in a signaling system to protect application-signaling entities from overload. A commonly known application-signaling protocol is Session Initiation Protocol (SIP) [11]. SIP allows us to establish, modify, or release

H. Tarasiuk (✉) · J. Rogowski
Institute of Telecommunications, Warsaw University of Technology,
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: halina@tele.pw.edu.pl

Jan Rogowski
e-mail: janek.rogowski@gmail.com

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_36

connections, e.g., for voice over IP applications. Therefore, we expect that a signaling system with SIP should offer the same level of overload control as in PSTN [4]. According to ITU-T recommendation [7], mean setup time for local calls should be not greater than 5 s and for international calls not greater than 8 s under normal network load conditions. On the one hand, signaling systems are provisioned for these normal load conditions; on the other hand, in real networks, there are periods of time when servers could be overloaded. We expect that even in overload conditions, new calls will still be accepted, and the call setup time will be according to requirements for PSTN networks. Moreover, we expect that the call acceptance rate should be close to the acceptance rate of systems under normal load conditions. However, it would be very difficult to obtain, since rejecting calls also requires some processing power.

There are many studies and activities, e.g., in IETF on overload control mechanisms in signaling systems with SIP. In [5], authors discuss models and design aspects of overload control. This document provides a good summary of the overload control mechanisms and the possibilities of their implementation considered in the literature. It also defines performance metrics. In our chapter, we follow notions assumed in this document.

Based on the current state-of-the-art technology, in our work,[1] we provide an extensive performance evaluation study on a loss-based (loss percentage) overload control mechanism. This mechanism seems to be the simplest one to implement in signaling systems [5]. The other mechanisms, such as rate-based [9, 10, 12], window-based [9, 12], and signal-based [2] require more complex processes in signaling entities, e.g., monitoring the number of sending entities. Moreover, these mechanisms provide customized fairness [12], which is rather not preferred in public networks. Loss-based overload control supports basic fairness; it means that each call in the system has the same chance of acceptance.

The main work related to our studies is [6]. In this chapter, the authors provide studies on two approaches to overload control mechanisms, hop-by-hop and end-to-end. Comparing with the above, we propose a simple non-preemptive priority queueuing scheme for handling overload control messages with the highest priority and other SIP messages with a lower priority. This scheme allows for the protection of SIP servers from handling signaling messages, which will be dropped, e.g., in the next SIP server. In our case, this priority queueing mechanism is also considered for both evaluated loss-based, overload control mechanisms hop-by-hop and end-to-end. In [6], the authors only compared end-to-end and hop-by-hop mechanisms in a simple bottleneck topology. In our chapter, we provide more detailed studies enhanced through more complex topologies. Moreover, in [6] the authors use a more complex priority scheme due to stateful implementation of SIP server and local overload control approaches. The authors assumed the lowest priority for local overload control. In our studies, loss-based overload control algorithms allow only 10 % load reduction in single feedback messages. Therefore, we argue that high

[1] This work is partially funded by Polish Ministry of Science and Higher Education, under contract number N N517 385838 “Modeling and analysis of signaling in QoS Internet”.

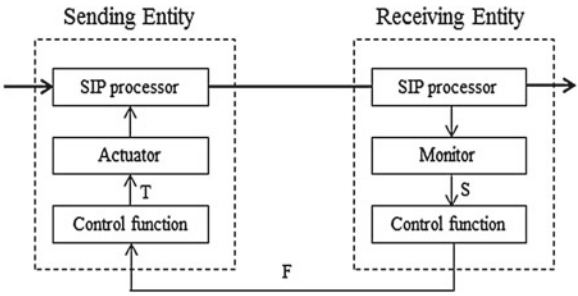


Fig. 1 General overload control system model

priority for handling overload control messages allows for faster reactions and thus prevents the server from overloading.

In [1] the authors provide the modeling of hysteretic signaling load control in NGN. However, they focus only on a simple FIFO queue with some thresholds for handling overload control messages.

The chapter is organized as follows: Section 2 describes the considered signaling system with SIP. Section 3 provides the results of performance evaluation studies. Section 4 concludes the paper.

2 Signaling System with SIP

Based on [5], we assumed a signaling system model with an overload control mechanism illustrated by Fig. 1. In this model, each two SIP servers (the sending entity and receiving entity) can be logically defined as following.

The SIP processor processes SIP messages and this component is protected from overload. Monitor, control function, and actuator are a part of the overload control layer. The monitor measures the SIP processor occupancy and sends sample S to the control function. The control function determines if an overload has occurred and whether a throttle T needs to be set to reduce the load. The control function on the receiving entity sends feedback F to the control function in the sending entity. In a loss-based mechanism, the control function on the sending entity may be omitted and feedback F can be used as throttle T for the actuator.

Figure 2 presents the assumed SIP model. In this model, end users are represented by User Agents (UAs). The session is established between the User Agent Client (UAC) and the User Agent Server (UAS). SIP messages typically traverse multiple SIP servers, whose main purpose is to route SIP messages. Initialization of a session begins with the UAC sending an INVITE message. UAS receives INVITE and responds with RINGING to indicate that the user has been alerted and OK to inform that the user has accepted the call. The UAC confirms receipt of OK by sending an ACK message, which ends the SIP three-way handshake. Call release is done by sending a BYE message, which is confirmed by an OK message. SIP uses 503-service unavailable messages to reject calls. When the SIP server receives an INVITE message and wants to reject that call, it simply drops the INVITE message and generates a 503-service unavailable message; this message is routed to the

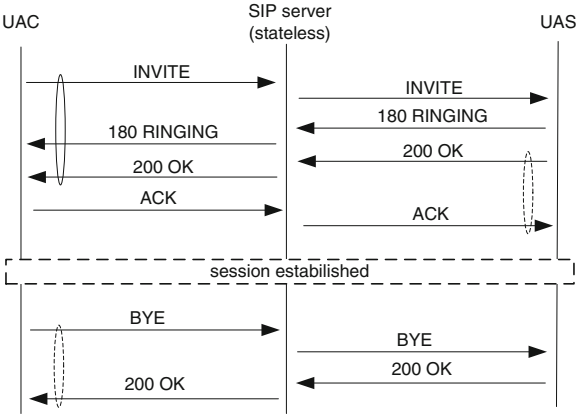


Fig. 2 SIP call flow

edge router and informs the user about the rejected call. We assume that the SIP servers are stateless, thus there is no 100-TRYING message in the call flow. We have implemented retransmission and timeout timers for invite and noninvite transactions (timers A, B, E, and F). In Fig. 2, invite transaction is marked with a continuous line and noninvite transactions are marked with a dotted line.

2.1 Overload Control Mechanisms

In a SIP signaling system, there is usually more than one SIP server on the path from source to destination. Thus, there are many ways of deploying the overload control components, monitor and actuator. In a hop-by-hop implementation, the monitor is located in the SIP server and corresponding actuator is in its direct downstream neighbor (see Fig. 3a). Therefore, there are independent overload control loops for each two neighboring SIP servers. Each SIP server throttles load that would otherwise overload its upstream neighbor. End-to-end implementation takes advantage of

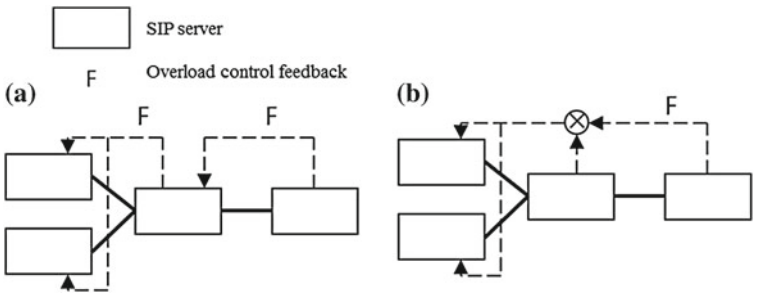


Fig. 3 **a**—hop-by-hop overload control, **b**—end-to-end overload control

multiple monitors providing overload control feedback to the actuator; there is an overload control loop along the entire path from source to destination. The point is to throttle the traffic as close to an ingress server as possible. Actuators are located at ingress servers and each SIP server provides overload control feedback to all actuators (see Fig. 3b). This allows it to throttle traffic on the first SIP server and not to use the server's resources to process messages that would be rejected.

The actuator can throttle traffic only if it knows that the request will eventually be forwarded to an overloaded server. Since routing is performed by each SIP server independently, the major difficulty is that the ingress server needs to know the whole path to the destination. One possible solution is to implement an end-to-end mechanism in a tightly controlled system with statically set routing [5].

For monitoring, we use an occupancy algorithm [6]. In each interval T_{m} the monitor measures server's SIP processor occupancy. It is averaged by the EWMA algorithm with weight w. The load reduce factor φ according to Eq. (1) is sent to the actuator each T_{f} interval.

$$\varphi = \frac{\rho_{\mathrm{target}}}{\rho_{\mathrm{m}}}, \tag{1}$$

where φ—load reduce factor, ρ_{target}—desired utilization, ρ_{m}—measured utilization. When the actuator receives feedback, it adjusts rejection factor p accordingly (2).

$$\mathrm{p} = \begin{cases} 1 \quad \text{when } p \cdot \varphi > 1 \\ p_{\min} \text{ when } p \cdot \varphi < p_{\min} \\ \quad p \cdot \varphi \text{ in other cases} \end{cases} \tag{2}$$

where p denotes INVITE rejection probability. Throttling requests are performed by a percent-thinning algorithm.

2.2 Model for SIP Server

We model the SIP server as a non-preemptive priority queueing (PQ) scheme with two finite queues with size B_1 and B_2, respectively, and a single server (see Fig. 4). The low-priority (LP) queue is dedicated for SIP messages and the high-priority (HP)

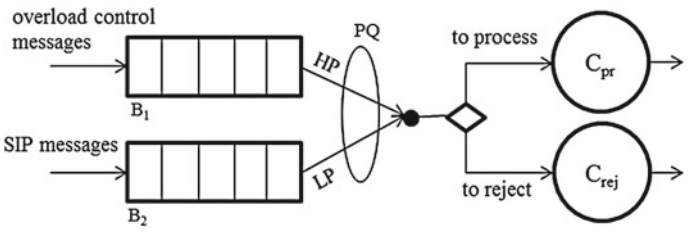


Fig. 4 SIP server model

queue is for overload control feedback messages. Both queues are served according to the FIFO discipline. The server has the same processing rate (C_{pr}) for both INVITE and non-INVITE messages. When the SIP server rejects a new request (INVITE message), it only takes a part of the time it would otherwise have taken to process this message (resolve SIP URI—Uniform Resource Identifier).

Rejecting rate (C_{rej}) is the number of messages per second a SIP server can reject by generating a 503-service unavailable message [11].

2.3 Model for UAs

Each new request is generated by a new instance of UAC. These requests are generated with a rate of λ requests per second according to the Poisson process. New requests are uniformly distributed over all MGs (media gateway) which aggregate UAs. SIP events were modeled based on [11]. The call holding time is the amount of time UAC waits until it begins call release. The call holding time is described by the exponential distribution with a mean value of 3 min.

3 Performance Evaluation

In this section, we present results for the goodput performance evaluation of the loss-based overload control mechanism. We investigate hop-by-hop and end-to-end performance in two complex topologies.

Based on the presented system model (see Sect. 2), we developed a discrete event simulator to evaluate performance metrics of SIP signaling systems. For this purpose, we define goodput as a total rate of calls per second (cps), which were terminated by users in the system (which completed whole sequence illustrated in Fig. 2). The offered load is the mean number of new requests per second that have arrived at the system.

We begin our performance evaluation studies with the topology illustrated in Fig. 5. It is the simplest topology with network diameter $D = 3$, in which the overload server can be two hops away from ingress servers. We assumed two ingress servers

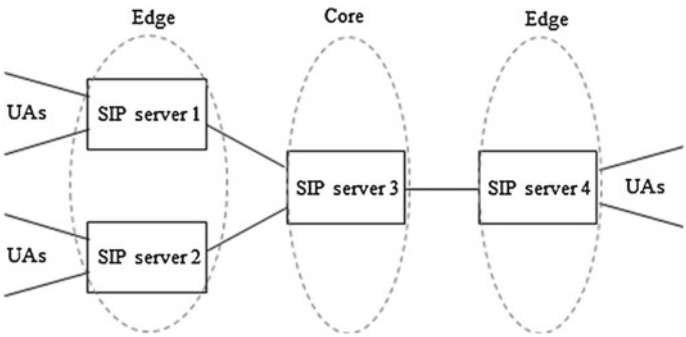


Fig. 5 Network topology 1

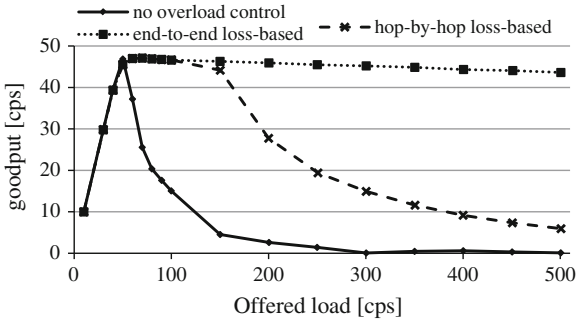


Fig. 6 Goodput as a function of offered load—network topology 1

to cause overload of SIP server 4 without, potentially, overloading the first two edge servers. In this scenario, we evaluate performance of a hop-by-hop approach. More precisely, the topology consists of 3 edge servers (SIP servers 1, 2, and 4) and one core server (SIP server 3). Edge servers have a processing rate of 300 messages per second and rejection rate of 1,200 messages per second, while the core server has processing rate of 480 messages per second and rejection rate of 1,500 messages per second. Buffers are set to $B_1 = B_2 = 60$ messages for all servers. Parameters for the loss-based algorithm were set to $T_f = 400$, $T_m = 200\,\text{ms}$, $w = 0.8$, $\rho_{\text{target}} = 0.95$. We chose to increase delay between feedbacks to decrease overhead caused by processing these messages (they are served with high priority). We increase weight in which new measurements are treated to compensate for that change.

Figure 6 illustrates evaluation results for a system with no overload control, hop-by-hop, and end-to-end loss-based overload control. One can see that the system with no overload control mechanism becomes congested as soon as the offered load reaches the server call processing rate (approx. 50 calls per second) resulting in a goodput drop. On the one hand, we may expect that hop-by-hop mechanisms generally perform well when the overloaded server is one hop away from the ingress server (e.g., if SIP server 3 would be overloaded). On the other hand, end-to-end mechanisms show their advantage in cases when the overload server is more than one hop away from the ingress server. Figure 6 shows that at a certain load, hop-by-hop goodput begins to drop. That is because servers 1 and 2 are not aware of overload of server 4 so they forward messages, which are going to be rejected anyway. Moreover, they process server 3's rejected messages (503-service unavailable) which also increases overhead. End-to-end mechanisms perform well at any load because new calls are rejected at ingress servers (as SIP server 1 and 2) which, by better resource usage, provides the best results.

Figure 7 presents network topology in which the next scenario has been evaluated. In this test, we show that the point at which hop-by-hop performance drops is indeed dependent on ingress servers' overload. We uniformly distributed traffic load between four edge servers and thus we caused less load to be handled by each server, which delayed goodput drop. Edge servers have a processing rate of 300 messages per

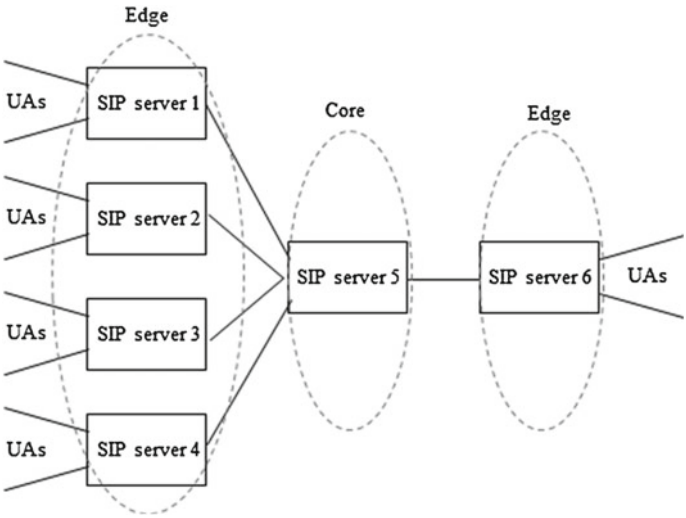


Fig. 7 Network topology 2

second and rejection rate of 1,200 messages per second, while the core server has a processing rate of 480 messages per second and rejection rate of 1,500 messages per second. Buffers are set to $B_1 = B_2 = 60$ messages for each server. Again, parameters for the loss-based algorithm were set to $T_f = 400$, $T_m = 200$ ms, $w = 0.8$, $\rho_{target} = 0.95$.

Figure 8 illustrates goodput characteristics for the system with no overload control, hop-by-hop and end-to-end loss-based overload control. Similar to the previous scenario, the system with no overload control mechanism becomes congested when offered load exceeds the server's processing rate (approx. 50 calls per second). Hop-by-hop and end-to-end mechanisms are on par until a certain point. This is when the ingress server becomes overloaded.

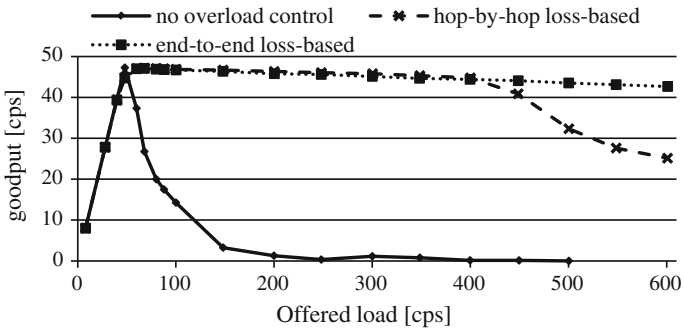


Fig. 8 Goodput as a function of offered load—network topology 2

That is because edge (ingress) servers process messages that are going to be rejected and additional overhead is added from processing rejected messages from the core server (which reject messages for overloaded server 6). With topology illustrated in Fig. 5 where there are only 2 ingress servers, the system becomes overloaded by a lesser value of offered load (see Fig. 6). This is because the offered load is distributed over 2 servers, which cause more overhead to each than it would cause if there were 4 ingress servers like in this scenario (see Fig. 7).

4 Summary

The paper provides comprehensive studies on loss-based overload control mechanisms in a signaling system with SIP. In the paper, we proposed a simple, non-preemptive priority queueing scheme for handling overload control messages with high priority in the signaling servers. Moreover, we compared two approaches of deploying overload control. They were hop-by-hop and end-to-end. We obtained our results for two complex topologies in which we demonstrated performances of the considered approaches. Based on simulation results obtained for the above-mentioned topologies, we conclude that a signaling system with a loss-based end-to-end overload control mechanism performs better than hop-by-hop and protects the system from overloading.

References

1. P. Abaev, Y. Gaidamaka, K.E. Samouylov, in *Modeling of Hysteretic Signaling Load Control in Next Generation Networks*, ed. by S. Andreev, et al. NEW2AN/ruSMART 2012. LNCS, vol. 7469 (Springer, Heidelberg, 2012), pp. 440–452
2. A. Abdelal, W. Matragi, Signal-based overload control for SIP servers, in *7th IEEE Consumer Communications and Networking Conference*, CCNC 2010 (IEEE Press, Las Vegas, 2010), pp. 1–7
3. T. Aoyama, A new generation network: beyond the internet and NGN, ITU-T kaleidoscope. IEEE Commun. Mag. (2009)
4. W. Berger, Comparison of call gapping and percent blocking for overload control in distributed switching systems and telecommunications networks. IEEE Trans. Commun. **39**(4), 574–580 (1991)
5. V. Hilt, E.C. Noel, C. Shen, A. Abdelal, Design considerations for session initiation protocol (SIP) overload control, in *Internet RFC 6357*, August 2011
6. V. Hilt, I. Widjaja, Controlling overload in networks of sip servers, in *16th IEEE International Conference on Network Protocols, ICNP 2008* (Orlando 2008)
7. ITU-Telecommunication Rec. E.671. Network Post-selection delay in PSTN/ISDN networks using Internet telephony for a portion of the connection, March 2000
8. ITU-Telecommunications Rec. Y.2012. Functional requirements and architecture of next generation networks, April 2010
9. E.C. Noel, C. Johnson, Novel overload controls for SIP networks, in *21st International Teletraffic Congress* (IEEE Press, Paris 2009), pp. 1–8

10. E.C. Noel, C.R. Johnson, *Initial simulation results that analyze sip based voip networks under overload*, ed. by L. Mason, T. Drwiega, J. Yan. in ITC 2007, LNCS, vol. 4516 (Springer, Heidelberg, 2007), pp. 54–64
11. J. Rosenberg et al., SIP: Session initiation protocol, in *Internet RFC 3261*, June 2002
12. C. Shen, H. Schulzrinne, E. Nahura, *Session Initiation Protocol (SIP) Server Overload Control: Design and Evaluation, IPTComm 2008*. LNCS, vol. 5310 (Springer, Heidelberg, 2008), pp. 149–173

Part VIII
Analysis of Physical, Social and Biological Systems

Improving Event Recognition Using Sparse PCA in the Context of London Twitter Data

Theo Pavlakou, Arta Babaee and Moez Draief

Abstract Motivated by some of the recent work based on using sparse principal component analysis to analyse social media, we propose an improvement which involves altering the input data matrices by considering what relationships they represent. Accordingly, we confirm our result by using Twitter data from London in the year 2012 as a medium to demonstrate on. Various alterations are made to the data matrix obtained from this data and the resulting matrices are then passed through a sparse principal component analysis algorithm. The resulting outputs are then analysed and it is shown that indeed the results do differ, with one particular variation consistently outperforming the rest. Our results are especially of interest when the data to be analysed can be represented by a binary matrix of some sort, e.g. in document analysis.

1 Introduction

Principal Component Analysis (PCA) is a mathematical technique concerned with finding linear combinations of the features in a set of features which lead to the highest variation in the outcomes, which leads to the highest information gain. The problem with PCA is that it can be hard to interpret its results since the size of the vector that is returned by it could be immense in length, with only a fraction of the entries being of significance. This has led to the creation of algorithms that return sparse representations approximating the principal components, which lead to results that are much easier to interpret. These will be referred to as Sparse PCA algorithms in this chapter.

Much work has been done in this area [1–4] and plenty others have created or improved existing algorithms to retrieve sparse principal component like vectors.

T. Pavlakou · A. Babaee (✉) · M. Draief
Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College London, London, UK
e-mail: arta.babaee@imperial.ac.uk

M. Draief
e-mail: m.draief@imperial.ac.uk

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_37

This chapter aims to focus on binary data and show that with knowledge of what the data represents beforehand, the data can be preprocessed in such a way that improves the output of these algorithms. The data used is a batch of Tweets from London throughout the year 2012.[1] The data was parsed and turned into a binary matrix, as is done in [6] and the same algorithm is used to be able to compare the results. The difference lies in the fact that the matrix to be given as an input to the algorithm will be changed. We will give reasons as to why the resulting matrix is a better choice as an input, for binary data matrices, with backing from empirical results using real data.

The structure of this chapter is as follows: in Sect. 3 the sparse principal component analysis algorithm that is used for the data analysis is explained. In Sect. 4 various modifications are made to improve the algorithm's performance. Testing is done on real Twitter data and the results are analysed. Finally, in Sect. 5 the results and analysis are discussed and further work is proposed.

2 Notation

Throughout the remaining sections, the following mathematical notations will be used. $\mathbf{S} \in \mathbb{Z}^{*m \times n}$ is the data matrix consisting of m data points (in the specific case considered in depth in this paper, the Tweets) evaluated on n features (words), with $[\mathbf{S}]_{i,j} = 1$ when Tweet i contains word j and zero otherwise. Likewise, the variable m is reserved for the number of Tweets and n for the number of words. $\mathbf{A} \in \mathbb{Z}^{*n \times n}$ is equal to $\mathbf{S}^T\mathbf{S}$ and this corresponds to the co-occurrence matrix of $\mathbf{S}$. $\mathbf{v}_i$ is to be used for the eigenvector with the ith largest eigenvalue, correspondingly denoted by λ_i. Both $\|\mathbf{x}\|$ and $\|\mathbf{x}\|_2$ denote the l_2 norm of a vector, defined as $\sqrt{\mathbf{x}^T\mathbf{x}}$ and $\|\mathbf{x}\|_0$ denotes the l_0-cardinality of a vector, defined as the number of non-zero elements in it. $[\mathbf{x}]_i$ denotes the ith element in a vector and similarly $[\mathbf{A}]_{i,j}$ denotes the element specified by the ith row and the jth column. For a matrix, $\|\mathbf{A}\|_F$ defines the Frobenius norm, defined as $\sqrt{\sum_{i=1}^{m}\sum_{j=0}^{n}[\mathbf{A}]_{i,j}^2}$.

3 Sparse PCA Through Low-Rank Approximation Algorithm

Sparse PCA algorithms aim to deliver similar results to PCA, but with the difference that they give rise to principal components that only have few non-zero entries. The reason sparsity is desired in the eigenvectors is because sparse approximations of the eigenvectors are useful for interpreting data that have large dimensions. The equation to be solved resolves to:

$$\mathbf{v}_1 = \underset{\|\mathbf{x}\|_2^2=1,\,\|\mathbf{x}\|_0=k}{\operatorname{argmax}} \left(\mathbf{x}^T\mathbf{A}\mathbf{x}\right), \tag{1}$$

[1] This is the same data which was used in [5].

where $\|\mathbf{x}\|_0$ is the l_0-cardinality of $\mathbf{x}$ i.e. how many non-zero elements $\mathbf{x}$ has. This is where Sparse PCA algorithms come in.

The algorithm used here to demonstrate the results is borrowed from [6] and is explained briefly in the following sections. It has been chosen as the authors of [6] use Twitter data to demonstrate its effectiveness and so, taking into account their results also, it provides a fair means for comparison.

3.1 Sparse PCA Algorithm

We assume that a rank d approximation of the co-occurence matrix, $\mathbf{A}_\mathrm{d}$, is arbitrarily close to the original based on the Forbenius norm. This can then be expressed as:

$$\mathbf{A}_\mathrm{d} = \sum_{i=1}^{d} \lambda_i \mathbf{v}_i \mathbf{v}_i^T.$$

The basic principle behind this algorithm is that the sparse approximation to the first principal component can be found by calculating the eigenvector $\mathbf{x}_*$ with the largest eigenvalue for all the possible k × k submatrices of $\mathbf{A}$. This then leads to finding $\binom{n}{k}$ possible sparse eigenvectors which would be quite wasteful since there will be a need to find $O\left(n^k\right)$ eigenvectors, where k is the sparsity of the eigenvector, i.e. how many elements of the vector are non-zero. In [6], it has been shown that only $\mathrm{O}\left(n^d\right)$ candidates need to be examined though and since $d < k$ typically, this means that the time complexity can be reduced significantly. This reduction is the result of the Spannogram algorithm which is also described by the authors and the resulting time complexity turns out to be $\mathrm{O}\left(n^{d+1}\log(n) + n^d k^2\right)$. Depending on the structure of the matrix $\mathbf{A}_\mathrm{d}$, either one of the terms could prevail.

3.2 The Spannogram Algorithm

The Spannogram algorithm aims to eliminate any candidates for the sparse eigenvectors which are redundant to calculate. This can be demonstrated by taking a Rank-1 matrix and solving the equation

$$\max_{\|\mathbf{x}\|_2^2=1, \|\mathbf{x}\|_0=k, \mathbf{x}\in\mathbb{S}_k} (\mathbf{x}^T \mathbf{A}_1 \mathbf{x}),$$

where $\mathbb{S}_k$ is the set of all k-sparse vectors in $\mathbb{Z}^{*n}$. The solution turns out to be:

$$\max_{\|\mathbf{x}\|_2^2=1,\|\mathbf{x}\|_0=k,\mathbf{x}\in\mathbb{S}_k} \lambda_1 \left(\mathbf{v}_1^T\mathbf{x}\right)^2 = \max_{\|\mathbf{x}\|_2^2=1,\|\mathbf{x}\|_0=k,\mathbf{x}\in\mathbb{S}_k} \lambda_1 \left(\sum_{i=1}^{n}[\mathbf{v}_1]_i[\mathbf{x}]_i\right)^2.$$

Therefore, it can be seen that just by keeping the k entries with the largest magnitude and making all the other entries equal to 0 in x, will satisfy the equation above and finding the solution turns out to be trivial.

For the case where $d > 1$, what the algorithm essentially does is to transform the problem into many Rank-1 case problems and solve them to find the best solution.

The eigenvectors of $\mathbf{A}_\mathrm{d}$, can be represented as,

$$\mathbf{V}_d = [\sqrt{\lambda_1}\mathbf{v}_1, \ldots, \sqrt{\lambda_d}\mathbf{v}_d],$$

$$\max_{\|\mathbf{x}\|_2^2=1,\|\mathbf{x}\|_0=k,\mathbf{x}\in\mathbb{S}_k} \mathbf{x}^T\mathbf{A}_d\mathbf{x} = \max_{\|\mathbf{x}\|_2^2=1,\|\mathbf{x}\|_0=k,\mathbf{x}\in\mathbb{S}_k} \|\mathbf{V}_d^T\mathbf{x}\|_2^2.$$

Consider a vector, $\mathbf{c} \in \mathbb{R}^d$ such that $\|\mathbf{c}\|_2 = 1$, which is a function of some vector ϕ and the elements in it are sinusoids taking as arguments the elements in ϕ, where each $\phi_i \in \left(\frac{-\pi}{2}, \frac{\pi}{2}\right]$. The equation is then transformed into a Rank-1 instance by creating the vector $\mathbf{u}_c = \mathbf{u}_d\mathbf{c}$ to give

$$\max_{\|\mathbf{c}\|_2=1} \left(\max_{\|\mathbf{x}\|_2^2=1,\|\mathbf{x}\|_0=k,\mathbf{x}\in\mathbb{S}_k} \left(\mathbf{v}_c^T\mathbf{x}\right)^2 \right). \tag{2}$$

When $\mathbf{c}$ is fixed there is a simple trivial solution to Eq. 2. Note that there is no actual need to calculate the actual value of the $\mathbf{v}_c$ for each ϕ to find the candidate indices for the k-sparse eigenvectors only the relative magnitudes between the different entries in $\mathbf{v}_c$ must be taken into account. The relative magnitudes will only change at intersection points between the sinusoids, and therefore, only for these points must a candidate eigenvector indices actually be tested. This reduces the computation to $O\left(n^d\right)$ instead of $O\left(n^k\right)$. A more thorough description and justification of the algorithm can be found in [6].

4 Improving the Performance of the Spannogram PCA Algorithm

In this section we show how we can further improve the Spannogram PCA algorithm by showing results using real-life data.

4.1 The Data Matrix: S

Let the matrix returned by the Twitter parser be $\mathbf{S} \in \mathbb{Z}^{*m \times n}$, where m is the number of tweets and n is the number of words in the bag of words. Typically, it is expected that in this application $m \gg n$, i.e. there are many more data points than the dimensionality of the data. For the matrices considered in this paper, $m \approx 40,000$ and $n = 3{,}000$. This is created almost identically to that in [6], i.e. words with less than 3 characters are omitted, common words such as "him" and "that" which convey little to no information are also filtered out.

4.2 The Co-Occurrence Matrix: A

A is the matrix that is of interest in this paper as it determines what sort of relationship the words appear to have with one another. This can be seen since the initial matrix is given by $\mathbf{A} = \mathbf{S}^T\mathbf{S}$ and can be viewed as the weighted adjacency matrix of an undirected graph with each vertex being a word and each edge weight being the total number of times that word appears with the word represented by the connecting vertex over all the Tweets. The element $[\mathbf{A}]_{i,j}$ is the total number of times word i appears with the word j in the set of Tweets being considered and any number on the diagonal, $[\mathbf{A}]_{i,i}$, is the total number of times word i appears.

In this section, variations of the matrix are considered and evaluated against each other. All testing is done by splitting the Twitter data into periods of roughly a day's worth of data, and this gives about 40,000 Tweets (m) and 3,000 words (n). The Spannogram Algorithm is run on each of the matrices using the same parameters in each case[2] and the first 8-sparse PC is computed.

The algorithm is tried on the initial co-occurrence matrix and 3 variants of it and the results are then analysed.

4.2.1 The Hollow Matrix: $\mathbf{A}_h$

In the initial matrix the diagonal represents the number of times each word appears in total, regardless of any relation to the other words i.e. each node has a link to itself, with the highest weighting. This means that words that occur very frequently have very large values on their diagonal, regardless of how they relate to other words, which can be misleading when taking the eigenvalues of the matrix, since words that may not have been filtered out of the bag of words, such as "haha" or "this", may appear very frequently but not have a distinct pattern as to the sorts of words they pair with most frequently. Since the whole point of the sparse principal components is to find a relationship of a few words together, it is only natural to want to prevent this. The matrix **A**, therefore, is substituted for $\mathbf{A}_h$ where for each of its elements,

[2] Using a rank 2 approximation.

$[\mathbf{A}_{\mathrm{h}}]_{i,j}$, it takes the value

$$[\mathbf{A}_h]_{i,j} = \begin{cases} [\mathbf{A}]_{i,j} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}. \tag{3}$$

4.2.2 The Weighted Laplacian Matrix: $\mathbf{A}_l$

The Laplacian of a graph is a useful representation of it and is very useful in applications involving Spectral Graph Theory and Partial Differential Equations. $\mathbf{A}_l$ of a graph is given by the following equation:

$$\mathbf{A}_l = \mathbf{D} - \mathbf{B}. \tag{4}$$

Here $\mathbf{D}$ is the degree matrix of the graph, which is a diagonal matrix and each element j on the diagonal is the sum of the weights of the edges that are connected to the vertex representing word j. $\mathbf{B}$ is the weighted adjacency matrix and is equal to $\mathbf{S}^T\mathbf{S}$. More about the Laplacian is explained in [9].

4.2.3 The Weighted Laplacian of the Hollow Matrix: $\mathbf{A}_{lh}$

This time the Laplacian matrix is created using the graph represented by the hollow matrix, $\mathbf{A}_h$ described above, i.e. there are no edges from a word to itself.

Table 1 The words associated with the first principal component of Twitter data during the period 23rd September 2012 to the 24th September 2012 using the Spannogram Algorithm on the variants of the co-occurrence matrix

Matrix $\mathbf{A}$		Matrix $\mathbf{A}_h$		Matrix $\mathbf{A}_l$		Matrix $\mathbf{A}_{lh}$	
Index	Word	Index	Word	Index	Word	Index	Word
1	haha	121	**Terry**	12	Them	5	Still
7	Yeah	120	**John**	68	Their	3	Night
3	Night	321	**International**	10	Great	7	Yeah
5	Still	908	**Retires**	44	Ever	12	Them
12	Them	772	**Retired**	33	Better	2	Need
6	Work	1558	**Retirement**	3	Night	11	Only
2	Need	66	**Football**	29	After	6	Work
9	Thanks	144	**England**	21	Weak	33	Better

Table 2 The words associated with the first principal component of Twitter data during the period 25th September 2012 to the 26th September 2012 using the Spannogram Algorithm on the variants of the co-occurrence matrix

Matrix **A**		Matrix $\mathbf{A}_h$		Matrix $\mathbf{A}_l$		Matrix $\mathbf{A}_{lh}$	
Index	Word	Index	Word	Index	Word	Index	Word
2	Need	12	**Please**	6	Them	6	Them
1	Haha	827	**Officers**	28	Over	4	Work
4	Work	889	**Murders**	31	Down	9	Only
3	Want	240	**Following**	32	After	2	Need
6	Them	783	**Fallen**	13	Make	13	Make
10	Still	756	**Recent**	42	Week	10	Still
7	Yeah	190	**Police**	43	Though	31	Down
9	Only	787	**Colleagues**	21	Look	67	Something

4.3 Results

The results can be seen in Tables 1, 2 and 3. As it can be seen, only the hollow matrix gives PCs which corresponds to an actual single distinct event that took place at the times considered. Table 1 refers to the retirement of England's former captain, John Terry, from international football.[3]

Table 2 is related to a murder that took place on the 19th September 2012, in which 2 police officers were killed[4] and resulted in a trending topic on Twitter. Finally, Table 3 emerges as a result of Europe's victory over the US in the Ryder Cup golf competition in 2012.[5]

Note that in the case of the matrix **A**, in every single occasion the words that appear are in the top 20 most frequently used words (as indicated by their index) and they do not really convey much information as to any particular event that took place in the periods considered. Essentially, the algorithm seems to be choosing these words purely because they occur very frequently. For instance, the word that is chosen for the matrix **A** that has the highest ranking index (and therefore the lowest frequency) in Table 3 is "only", which appears 541 times. Similar results can be seen for the matrices $\mathbf{A}_l$ and $\mathbf{A}_{lh}$, though not as extreme, and are all consequences of the fact that the diagonal elements always overpower the off-diagonal ones in the resulting matrices. On the contrary, for the case of the hollow matrix, the lowest frequency word is "teameurope", which appears merely 160 times and for other PCs this is even lower ("retirement" has an index of 1558).

[3] Confirmed by the Guardian: http://www.theguardian.com/football/2012/sep/23/john-terry-retires-international-football.

[4] Confirmed by the BBC: http://www.bbc.co.uk/news/uk-england-19634164.

[5] Confirmed by the BBC: http://www.bbc.co.uk/sport/0/golf/19780678.

Table 3 The words associated with the first principal component of Twitter data during the period 30th September 2012 to the 1st October 2012 using the Spannogram Algorithm on the variants of the co-occurrence matrix

Matrix **A**		Matrix $\mathbf{A}_h$		Matrix $\mathbf{A}_l$		Matrix $\mathbf{A}_{lh}$	
Index	Word	Index	Word	Index	Word	Index	Word
2	Haha	14	**rydercup**	16	Them	7	Still
4	Need	38	**Europe**	9	Only	6	Xfactor
8	Yeah	91	**rydercup2012**	7	Still	8	Work
5	Work	160	**teameurope**	44	Year	27	Right
7	Still	12	**Great**	27	Right	40	Life
6	Xfactor	77	**Golf**	64	Weekend	112	Other
3	Night	23	Done	37	Week	162	There's
9	Only	102	**Ryder**	8	Work	9	Only

5 Conclusion

We showed that using Sparse PCA algorithms it is possible to infer results from microblogs such as Twitter, as is done both in [5, 6]. Different matrices representing the data features have then been analysed and fed to the algorithm for Sparse PCA to produce results. These have shown that the hollow matrix, $\mathbf{A}_h$, gives the best results by far when comparing the resulting principal components returned for each of the inputs. Moreover, though the Twitter Data is the same as that used in [5], the events that are discovered are different but still completely valid as can be confirmed by other sources. This highlights the fact that this approach may be well suited in conjunction to the methods described here. The choice of this matrix, as opposed to the one used in [6], can be applied to any sparse principal component analysis application when using binary data to discover relationships in features that may otherwise be overpowered by components that do not necessarily give the best results.

Acknowledgments This work has been carried out in the scope of the EC funded project SMART (FP7-287583).

References

1. H. Shen, J.Z. Huang, Sparse principal component analysis via regularized low rank matrix approximation. J. Multivar. Anal. **99**, 1015–1034 (2007)
2. A. d'Aspremont, L.E. Ghaoui, M. I. Jordan, G.R.G. Lanckriet, A direct formulation for sparse pca using semidefinite programming, in NIPS (2004)
3. H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis. J. Comput. Graph. Stat. **15**, 2006 (2004)
4. X.-T. Yuan, T. Zhang, Truncated power method for sparse eigenvalue problems, ArXiv e-prints, Dec 2011

5. M.-D. Albakour, C. Macdonald, I. Ounis, Identifying local events by using microblogs as social sensors, in Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, OAIR '13, (Paris, France, France), LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2013, pp. 173–180
6. D.S. Papailiopoulos, A.G. Dimakis, Sparse pca through low-rank approximations, (2013)
7. L. Mackey, Deflation methods for sparse pca, in *NIPS*, ed. by D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Curran Associates Inc, New York, 2008), pp. 1017–1024
8. M. Journée, Y. Nesterov, P. Richtárik, R. Sepulchre, Generalized power method for sparse principal component analysis. J. Mach. Learn. Res. **11**, 517–553 Mar. 2010
9. D.A. Spielman, Algorithms, graph theory, and the solution of laplacian linear equations. ICALP **2**, 24–26 (2012)

Modeling Structural Protein Interaction Networks for Betweenness Analysis

Deniz Demircioğlu, Özlem Keskin and Attila Gursoy

Abstract Protein–protein interactions are usually represented as interaction networks (graphs), where the proteins are represented as nodes and the connections between the interacting proteins are shown as edges. Proteins or interactions with high betweenness are considered as key connector members of the network. The interactions of a protein are dictated by its structure. In this study, we propose a new protein interaction network model taking structures of proteins into consideration. With this model, it is possible to reveal simultaneous and mutually exclusive interactions of a protein. Effect of mutually exclusive interactions on information flow in a network is studied with weighted edge betweenness analysis and it is observed that a total of 68 % of bottlenecks found in p53 pathway network differed from bottlenecks found via regular edge betweenness analysis. The new network model favored the proteins which have regulatory roles and take part in cell cycle and molecular functions like protein binding, transcription factor binding, and kinase activity.

Keywords Bioinformatics · Protein–protein interaction · Protein–protein interaction network · Betweenness analysis

1 Introduction

Proteins are important biological macromolecules due to their participation in many different biological processes either with other proteins or other molecules such as DNAs, RNAs, lipids, etc. They take part in these reactions by interacting with these

D. Demircioğlu · A. Gursoy (✉)
Department of Computer Engineering, Koç University, Sariyer, 34450 İstanbul, Turkey
e-mail: agursoy@ku.edu.tr

D. Demircioğlu
e-mail: ddemircioglu@ku.edu.tr

Ö. Keskin
Department of Chemical and Biological Engineering, Koç University, Sariyer,
34450 İstanbul, Turkey
e-mail: okeskin@ku.edu.tr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_38

molecules in the form of complexes either with proteins or other molecules. Due to their importance for biological processes, protein–protein interactions are studied extensively both with low throughput experiments on the scale of a few specific interactions and high throughput on the scale of hundreds of interactions thanks to experimental techniques such as yeast two hybrid screens. The extensive interaction data obtained from these studies allowed the researchers to construct protein–protein interaction (PPI) networks which proved to be useful for determining biologically and functionally important proteins and modules [1, 2]. In PPI network representation, nodes correspond to proteins and an edge between two nodes implies an interaction between those two proteins. More formally a PPI network is a graph $G = (V, E)$ where V is a set of proteins and E is a set of edges that gives information about the interaction of the two nodes which it connects.

Abstraction of data allowed the researchers to analyze the topological properties of biological networks such as clustering coefficients, shortest path algorithms, centrality calculations, etc. The analysis of PPI networks revealed that they are scale-free networks which reshaped the approach to the PPI networks and their analysis [3]. Scale-free networks are resistant to random node failures. Even if a significant number of randomly selected nodes are removed from the network, the remaining nodes will still be connected [4]. On the other hand, scale-free networks are vulnerable to targeted attacks to the hubs [5]. Several studies focusing on the betweenness in PPI networks shed light on a non-hub centric organization (co-ordination of modules by non-hub proteins) of the protein interaction networks and in a way supported the idea that some interactions are more important in the network structure. Valente and Cusick characterized the yeast proteome using a betweenness centrality [6]. The coordinated functionality of the yeast interactome is likely to be achieved not only by central proteins but also by the pairwise interactions of with high traffic nodes (i.e., they have high betweenness scores) mediating coordination across modules.

In all these studies, the interactions of a protein are assumed to be happening at the same time. However, interactions of a protein are dictated by its structure. We propose a new protein interaction network model taking structures of proteins into consideration which is more realistic. With the addition of structural information, it is possible to reveal simultaneous and mutually exclusive interactions of a protein [7, 8]. The new network model incorporates the simultaneous/exclusive interaction information as edge weights. In this chapter, first we describe how to augment a typical PPI network with structures and model it with the simultaneous/exclusive interaction information. Then, using p53 pathway network as an example, an important tumor suppressor signaling network in cells, we present the comparison of betweenness analysis of the typical p53 PPI and the new p53 PPI network. We believe that the more realistic representation of the PPI networks leads to biologically more relevant conclusions. This representation may result in more accurate remarks when used for topological network analysis algorithms such as network alignment algorithms [9].

2 Materials Methods

2.1 Structural Interaction Networks

In most network studies, protein–protein interactions are abstracted with a graph representation where proteins are represented as nodes and protein–protein interactions as edges. In order to incorporate the information on how proteins interact with each other, we first need to determine three-dimensional structural information of two interacting proteins in complex with each other. This is a major challenge in structural biology. The protein data bank (PDB) [10] contains the structure information of some interactions, but the majority of the interactions need to be predicted. Our group has developed a computational method, PRISM (Protein Interaction by Structural Matching) algorithm, to predict structural model of protein interactions previously unknown [11]. PRISM uses the conservation of the structural properties of protein molecules thorough the evolution. It is known that proteins interact with each other thorough regions called interfaces (complementary binding sites on their surfaces) which are structurally conserved through the course of evolution compared to the other regions. This conservation allows the prediction of new protein complexes by using their structural similarity to an existing protein binding site.

In this study, PRISM algorithm uses this principle to predict new protein complexes. PRISM is used to determine the interacting residues on the surface of the proteins which interacts with each other. A protein can interact with other proteins by using either the same region or different regions on its surface. If a protein uses the same region to interact with two different partners, these partners will be in a competing position to bind to the target protein. This introduces a limiting factor to the interactions and provides a valuable information for the analysis of PPI. We proposed to integrate these aspects of the PPI into the network by determining the overlapping interactions and embedding this overlap information in the form edge weights (weight represents the number of competing interaction partners). Hence the structural interaction network is an undirected weighted graph $G = (V, E)$ where V is a set of proteins and E is the set of edges where each edge e is associated with integer $w(e)$ weight.

2.2 Definition of Overlapping Binding Sites

A binding site is defined as overlapping if at least one of the residues of the binding site is used in the interaction with at least two interacting partners. For example in Fig. 1, CDKN2D interacts with CHEK1 protein through interface (contacting binding sites of two interacting proteins) 1bi7AB and with CDKN1B protein through interface 1jsuBC. The residues of 1bi7AB interface on CDKN2D protein are:

G.14 **A.17 R.18** G.19 **R.40 F.41K.43 V.48** M.50 S.53 T.84 F.86 E.117 H.119.

And the residues of 1jsuBC interface on CDKN2D are:

D.10 **A.17 R.18** D.20 E.23 R.26 L.27 R.30 D.31 N.39 **R.40 F.41 K.43** Q.47 **V.48** D.71 T.75 D.80.

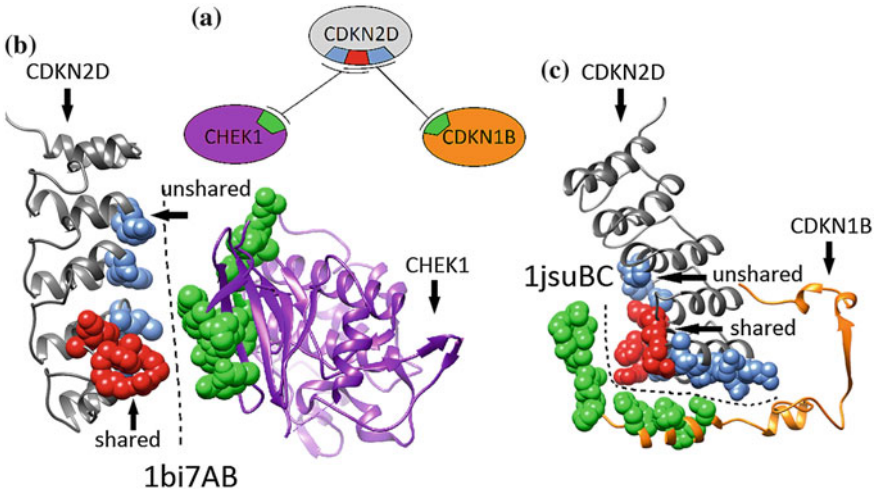


Fig. 1 Cartoon representation and 3D structures of proteins CDKN2D (*black*), CHEK1 (*magenta*) and CDKN1B (*orange*) and PRISM predicted binding sites on them (*spheres*). **a** Cartoon representation of the overlapping interaction. Same coloring scheme is used for both the cartoon and 3D structure representation. **b** Interface that CDKN2D (*black*) uses to interact with CHEK1 (*magenta*) is shown as *red* and *blue* where *red* residues are overlapping residues with another interface on CDKN2D. **c** Interface that CDKN2D (*black*) uses to interact with CDKN1B (*orange*) is shown as *red* and *blue* where *red* residues are overlapping residues with another interface on CDKN2D

The residues marked as bold are common in both binding sites, thus if one interaction occupies them other interaction can not happen. Overlapping of 1bi7AB and 1jsuBC interfaces on CDKN2D can be seen in Fig. 1 drawn by using UCSF Chimera [19]. In Fig. 1a the cartoon representation of CDKN2D with CHEK1 and CDKN1B is shown and in Fig. 1b, c the residues causing the overlap can be seen.

2.3 P53 Pathway Structural Network

P53 interaction network data is gathered from the literature [8, 12]. The structural model of interactions is computed by using PRISM. There were 81 nodes and 240 edges that form the p53 pathway network. Of the 240 edges, PRISM predicted the structure of 131 interactions. For 88 interactions there were no predictions found by PRISM. Hence the final network had 81 nodes and 240 edges, for 131 of which the interacting residues on the surface is determined. In Fig. 2a subnetwork of p53 is shown as an example. In part (a) the classical representation is shown with all edge weights are 1. In part (b) the edge weights calculated due to overlap is shown. For example, using the PRISM results, the weight of SKP2—CDK5 is computed as follows: The edge weight is 1 initially. SKP2—CDK5 interaction and SKP2—RAF1 interaction shares the same binding site on SKP2 its weight is increased by 1. Since some of these residues are also being used by another interaction (not shown here)

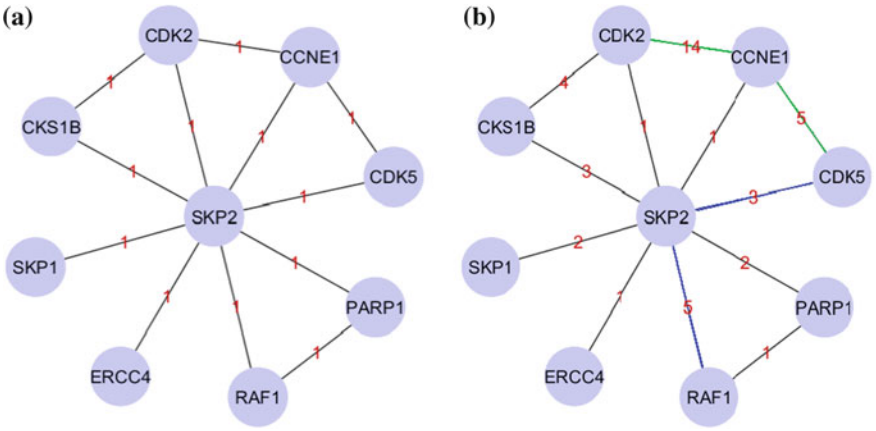


Fig. 2 **a** Weights of edges in a PPI network without taking overlap information into consideration. **b** Weights of edges determined based on mutually exclusive interactions

the weight of the edge is increased to 3. Same procedure applied to all overlapping edges to determine the final edge weights.

For overlapping interactions (binding sites having common residues), the number of different targets that binds to the same region is stored as the weight of the edge in the structural interaction network. The resulting network is visualized using Cytoscape [13] and can be seen in Fig. 3.

2.4 Definition of Hubs and Bottlenecks

Hub proteins in protein–protein interaction networks are defined as highly connected nodes, typically top 20 % of the highest degree nodes. Bottleneck edges are determined based on edge betweenness centrality and similarly they correspond to the top 20 % of edges with the highest edge betweenness centrality [14]. Betweenness centrality calculations are done using NetworkX [15] package for Python. Algorithm is defined by Ulrik Brandes such that: edge betweenness centrality of an edge is the sum of the fraction of all-pairs' shortest paths that pass through this particular edge [16]. Edge betweenness centrality is calculated as:

$$c_b(e) = \sum_{s,t \in V} \frac{\sigma(s,t|e)}{\sigma(s,t)} \tag{1}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest (s,t)-paths, and $\sigma(s,t|e)$ is the number of those paths passing through edge e.

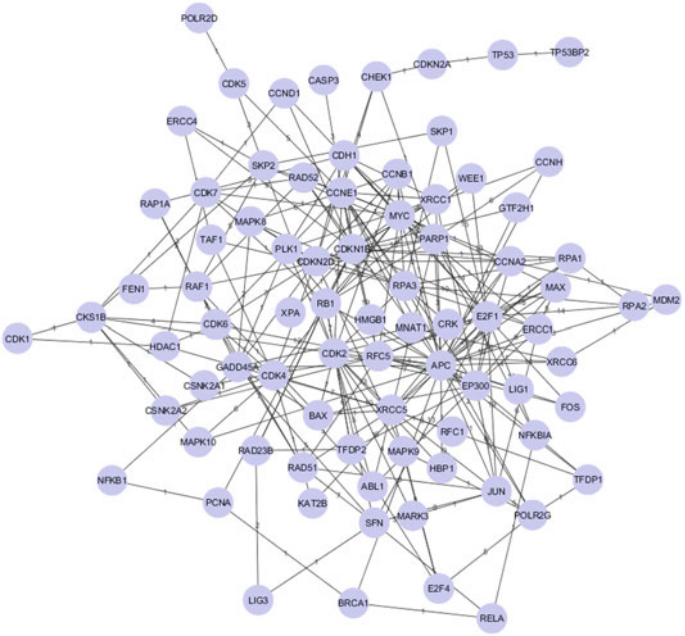


Fig. 3 The p53 network containing 81 genes and 240 interaction edges

3 Results

In order to see the effect of interfaces on an interaction network, edge betweenness centrality analysis is first done on the p53 prediction network and the top 50 edges (∼20 % of the total number (240) of edges) are labeled as bottlenecks. A weighted betweenness centrality analysis is done as explained in methods part. Bottleneck edges are determined by taking 50 edges (∼20 % of the total number of edges) with highest edge betweenness values.

3.1 Hub Nodes of P53 Network

For both network types (weighted and unweighted), hub nodes are identified. In total out of the 16 hubs, 3 (18.75 %) genes are identified as hubs by only one of the methods. When these different hub nodes are further examined, it is observed that one of the hubs in the weighted network is RB1 proto-oncogene. RB1 which codes the retinoblastoma protein is known to be dysfunctional in several major cancers [17].

3.2 Bottleneck Analysis of Weighted Network

Bottleneck edges found by analyzing structural interaction networks are 68 % (34 interactions) different from the bottleneck edges found by unweighted edge betweenness analysis. With this analysis some proteins became significant in bottleneck edges whereas they were absent in bottlenecks of control analysis. Between two different edge betweenness calculations the importance of interactions between some proteins changes. The proteins that are at the ends of the bottleneck edges are further investigated to assess the reliability of the proposed network model. In total there were 47 nodes (genes) that are found to be at the end of the bottleneck edges for the unweighted network and 42 for the weighted network.

In order to deduce the reason for this change, GO Biological Process and Molecular Function Annotation of genes coding proteins, which take part in the ends of the bottleneck edges, are examined. In Table. 1, the Biological Process Annotations of proteins that take part in bottlenecks for these two different network models are

Table 1 GO Biological Process Annotations of genes that were identified as critical by different betweenness analysis

Critical in Weighted			Critical in Unweighted		
GO ID	Term	P-value	GO ID	Term	P-value
GO:0051726	Regulation of cell cycle	4.3542E-21	GO:0006974	Response to DNA damage stimulus	1.0295E-28
GO:0007049	Cell cycle	1.5880E-18	GO:0051716	Cellular response to stimulus	3.3404E-25
GO:0006974	Response to DNA damage stimulus	1.2000E-17	GO:0033554	Cellular response to stress	3.7582E-25
GO:0033554	Cellular response to stress	8.2753E-17	GO:0006281	DNA repair	5.1133E-25
GO:0022402	Cell cycle process	5.3170E-16	GO:0006259	DNA metabolic process	3.5800E-24
GO:0051716	Cellular response to stimulus	3.0975E-15	GO:00051726	Regulation of cell cycle	1.1128E-22
GO:0051329	Interphase of mitotic cell cycle	4.9726E-15	GO:0006289	Nucleotide—excision repair	2.2589E-17
GO:0051325	Interphase	7.6976E-15	GO:0000079	Regulation of cyclin-dependent protein kinase activity	4.1665E-17
GO:0006259	DNA metabolic process	2.6117E-14	GO:0090304	Nucleic acid metabolic process	1.4177E-16
GO:0022403	Cell cycle phase	3.2878E-14	GO:0006950	Response to stress	1.6388E-16

The biological process is found for the nodes (genes) at the ends of the bottleneck edges for both analysis methods

listed. The common biological functions of those genes that are identified as critical in weighted edge betweenness analysis are more related to cell cycle compared to the genes found by unweighted edge betweenness analysis. The unweighted edge betweenness analysis identified genes related to DNA repair and stimulus response as critical.

In Table. 2, the Molecular Function Annotations of proteins, that take part in the ends of the bottleneck edges for both analysis, are listed. The common molecular functions of genes identified as critical for weighted edge betweenness analysis are transcription activation and kinase activity. For the unweighted network, critical nodes have mostly kinase activity as molecular function. These analysis are performed by BiNGO plugin [18] of the Cytoscape and p-values are obtained by performing hypergeometric testing.

Table 2 GO Molecular Function Annotations of genes that were not found at the bottleneck nodes in unweighted betweenness centrality analysis but found in weighted betweenness centrality analysis

Critical in Weighted			Critical in Unweighted		
GO ID	Term	P-value	GO ID	Term	P-value
GO:0005515	Protein binding	1.8086E-12	GO:0005515	Protein binding	3.1513E-12
GO:0004674	Protein serine/threonine kinase activity	6.6475E-10	GO:0016538	Cyclin—dependent protein kinase regulator activity	1.0321E-11
GO:0004672	Protein kinase activity	1.8862E-9	GO:0030291	Protein serine/threonine kinase inhibitor activity	1.0321E-11
GO:0016772	Transferase activity, transferring phosphorus—containing groups	4.2675E-9	GO:0004861	Cyclin—dependent protein kinase inhibitor activity	9.5698E-11
GO:0008134	Transcription factor binding	8.6146E-9	GO:0019887	Protein kinase regulator activity	1.9245E-10
GO:0016773	Phosphotransferase activity, alcohol group as acceptor	1.6654E-8	GO:0043566	Structure—specific DNA binding	4.1870E-10
GO:0016301	Kinase activity	4.5201E-8	GO:0019207	Kinase regulator activity	5.0560E-10
GO:0016563	Transcription activator activity	1.6385E-7	GO:0004860	Protein kinase inhibitor activity	1.2377E-9
GO:0003677	DNA binding	6.0298E-7	GO:0019210	Kinase inhibitor activity	1.7292E-9
GO:0004693	Cyclin—dependent protein kinase activity	1.0688E-6	GO:0047485	Protein N—terminus binding	1.0649E-7

4 Conclusion

In this study with the use of interface information coming from PDB structures of proteins and PRISM predictions, a more realistic network model for protein–protein interactions representing simultaneous mutually exclusive interactions has been proposed. Structural information clarified the competition between interactions such that some interactions will be in a race with each other if they are using the same binding site on a particular protein. These dependencies are expected to have an impact on signal transduction and data flow through individual proteins. Topological analysis of the new (weighted) network model provided further results relating network topology to biological interpretation compared to typical PPI network. Hub node analysis of the weighted network revealed an important proto-oncogene. The edge betweenness analysis of weighted network identified proteins with cell cycle related functions as critical nodes.

Acknowledgments Deniz Demircioğlu is supported by a TÜBİTAK (The Scientific and Technological Research Council of Turkey) fellowship. This work has been partially supported by TÜBİTAK, Research Grant Number: 113E164.

References

1. A.L. Barabási, Z.N. Oltvai, Network biology: understanding the cell's functional organization. Nat. Rev. Genet. **5**(2), 101–113 (2004)
2. H. Yu, D. Greenbaum, H. Xin Lu, X. Zhu, M. Gerstein, Genomic analysis of essentiality within protein networks. Trends Genet. **20**(6), 227–231 (2004)
3. A.L. Barabási, R. Albert, Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)
4. R. Albert, H. Jeong, A.L. Barabási, Error and attack tolerance of complex networks. Nature **406**(6794), 378–382 (2000)
5. H. Jeong, S.P. Mason, A.-L. Barabási, Z.N. Oltvai, Lethality and centrality in protein networks. Nature **411**(6833), 41–42 (2001)
6. A.X. Valente, M.E. Cusick, Yeast Protein Interactome topology provides framework for coordinated-functionality. Nucleic. Acids Res. **34**(9), 2812–2819 (2006)
7. A. Gursoy, O. Keskin, R. Nussinov, Topological properties of protein interaction networks from a structural perspective. Biochem. Soc. Trans. **36**(Pt 6), 1398–1403 (2008)
8. N. Tuncbag, G. Kar, A. Gursoy, O. Keskin, R. Nussinov, Towards inferring time dimensionality in protein–protein interaction networks by integrating structures: the p53 example. Mol. Biosyst. **5**(12), 1770–1778 (2009)
9. H.T.T. Phan, M.J.E. Stemberg, E. Gelenbe: Aligning protein-protein interaction networks using random neural networks. in: *Proceedings of the Bioinformatics and Biomedicine* (BIBM), 2012 IEEE International Conference on, 1–6 (2012)
10. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The protein data bank. Nucleic Acids Res. **28**(1), 235–242 (2000)
11. N. Tuncbag, A. Gursoy, R. Nussinov, O. Keskin, Predicting protein–protein interactions on a proteome scale by matching evolutionary and structural similarities at interfaces using PRISM. Nat. Protoc. **6**(9), 1341–1354 (2011)
12. K.W. Kohn, Molecular interaction map of the mammalian cell cycle control and DNA repair systems. Mol. Biol. Cell. **10**(8), 2703–2734 (1999)

13. P. Shannon, A. Markiel, O. Ozier, N.S. Baliga, J.T. Wang, D. Ramage, N. Amin, B. Schwikowski, T. Ideker, Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome. Res. **13**(11), 2498–2504 (2003)
14. H. Yu, P.M. Kim, E. Sprecher, V. Trifonov, M. Gerstein, The importance of bottlenecks in protein networks: correlation with gene essentiality and expression dynamics. PLoS Comput. Biol. **3**(4), e59 (2007)
15. A.A. Hagberg, D.A. Schult, P.J. Swart: Exploring network structure, dynamics, and function using NetworkX. in: *Proceedings of the 7th Python in Science Conference* (SciPy2008), 11–15 (2008)
16. U. Brandes, On variants of shortest-path betweenness centrality and their generic computation. Social Netw. **30**(2), 136–145 (2008)
17. A.L. Murphree, W.F. Benedict, Retinoblastoma: clues to human oncogenesis. Science **223**(4640), 1028–1033 (1984)
18. S. Maere, K. Heymans, M. Kuiper, BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. Bioinformatics **21**(16), 3448–3449 (2005)
19. E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, T.E. Ferrin, UCSF Chimera-a visualization system for exploratory research and analysis. J. Comput. Chem. **25**(13), 1605–1612 (2004)

Three-Dimensional Haptic Manipulator Controlled Game in the Treatment of Developmental Coordination Disorder

Anna Wałach and Agnieszka Szczęsna

Abstract Computer games are more and more used for serious purposes not only for entertainment. In few last years, we can observe very fast market development of serious games. They are used to education, simulation, research, health, and therapy purposes. This article describes a game controlled by Phantom Omni device that could be a rehabilitation tool in the treatment of developmental coordination disorder. The game focuses on usage of the feedback generated by the haptic device.

Keywords Serious game · Developmental coordination disorder · Haptic manipulator

1 Introduction

Specialized devices are being developed to resemble real-world reaction and be able to precisely identify operator's movements and intentions. The kind that focuses on generating real-time responses is called haptic devices. The most important feature of this type of controller is a tactile feedback produced by it that render the feeling of the texture and shape of objects shown on the screen of the computer.

Haptic devices are widely used in medicine. Haptic devices using virtual reality have been suggested to enhance the stroke rehabilitation. Research implies that haptic-based virtual rehabilitation offers a potential tool in motor and cognitive rehabilitation, with a wide range of applicability [1, 2].

Studies show increasing number of neurodevelopmental disorders being diagnosed among children and adults. One of them is dyspraxia, also known as developmental coordination disorder (DCD), an inability to plan, organize, and coordinate movement [3]. Depending on the level of severity, it can be a reason of small difficulties in everyday activities, for example fastening buttons or shoelaces, but may also make one unable to drive a car because of affecting working memory and coordination skills.

A. Wałach · A. Szczęsna (✉)
Institute of Informatics, Silesian Technical University, Akademicka 16, 44–100 Gliwice, Poland
e-mail: agnieszka.szczesna@polsl.pl

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_39

There are several various difficulties that occur during treatment of DCD. One of the most significant obstacles is a problem with conducting proper diagnose. Another existing system of therapy, not all of them equally effective, but all requiring experts in rehabilitation or psychology and hours of exercises with professionals.

DCD is very often observed among children in the primary school. The most efficient way of engagement into the exercise this age group seems to be the serious game. It allows to develop the necessary coordination or mental skills. Disguising the exercise as a game makes it more entertaining and pleasurable for a child.

Serious game is a general term used for application that is developed using a computer game technology and game design principles but are used for non-entertainment purposes [4]. We can say that these applications are any sort of interactive computer-based game software for one or multiple players to be used on any platform and that has been developed with the intention to be more than entertainment [5]. The idea is such games could be used for more serious purposes such as education, simulating real-world phenomenon and relations, increasing life quality through health, rehabilitation, and therapy applications or raising interest to the problems in our global world [6]. The main point of serious games is: to get players to learn something and, if possible, have fun doing it. Enjoyment increases the player's engagement. One of main group of serious games are games for health [7–10]. Games are rapidly becoming an important tool for improving health behaviors ranging from healthy lifestyle habits and behavior modification, to self-management of illness and chronic conditions to motivating and supporting physical activity. The findings also suggest that play and entertainment can be effective foundations for serious interventions in health care.

This article describes a coordination improving serious game, controlled by three-dimensional haptic device that could be an effective tool in rehabilitation of children with dyspraxia.

2 Developmental Dyspraxia

Developmental Coordination Disorder (DCD), also known as developmental dyspraxia, is the inability to plan, organize and coordinate movement [3, 11–14]. It results in fine and gross motor problems and/or speech difficulties, which can have strong negative impact on daily activities and school achievements. Dyspraxia was first described by WHO in 1992. It is a hard to diagnose and even harder to cure motor disorder, which is subcategory of neurodevelopmental disorder—along with autism and Down syndrome—and mostly identified among children. It often appears together with neurological illnesses from other subcategories, like ADHD, Asperger syndrome, or dyslexia [15]. It is, however, also sporadically detected in completely healthy (except for DCD) children who are developing well intellectually. DCD is not caused by motor or sensory impairments, similar to muscular dystrophy or Parkinson's disease. No correlation to any known neurological condition or intellectual impairment could be found.

2.1 Treatment

The treatment for DCD is based on rehabilitation and training. No pharmacological therapy was designed to be used during DCD's treatment. Medications are reserved for the treatment of associated conditions (e.g., attention deficit hyperactivity disorder, ADHD) [16].

There are two basic approaches of rehabilitation procedure: top-down and bottom-up. The top-down approach, also known as task-oriented or modular, attempts to remedy or improve specific difficulties by employing specific techniques aimed at the observed motor challenge (e.g., difficulty with handwriting, catching a ball, or performing fine motor tasks with the fingers). It usually involves gradually targeting certain problem behaviors and implementing step-by-step interventions that focus on teaching and practicing the skill [16]. The most important element of modular-approach is practice. Practical applications of this method are therapies such as the cognitive motor approach with task orientation or task-oriented approaches with motor learning.

The bottom-up approach (process-oriented, deficit-oriented), the more global or generalized one is based on the theoretical assumption that the motor skills problem is just a manifestation of some underlying mechanism, such as impaired sensory integration or insufficient or inaccurate kinesthetic perceptions. In the bottom-up approach, the therapist does not initially address the observable motor challenge. Rather, the expert focuses on how children manage their bodies, process stimulation (sensory information), and deal with problems. The expectation is that the improved sensory-motor (sensorimotor) functioning becomes generalized and eventually improves the motor skills. As children become comfortable with their bodies, they gain control of their motor (and other) functions [16]. This method is represented by the kinesthetic training approach, sensorimotor integration therapy and sensory integration therapy.

No researches show that any of this approaches is definitely effective as DCD's treatment. This is probably caused by great diversity in symptoms and severity's levels of DCD. However, data suggest that top-down approach may be more efficient than bottom-up.

3 Haptic Devices in Rehabilitation

In this particular area there has been great interest in research how computer haptics technology can assist in rehabilitation process of people who has had some damage to their motor skills, while playing a game. The combination of the properties inherent to computer haptics (force and tactile feedback) with the appealing and motivating factor offered by games with virtual reality have provided a framework for the development of various rehabilitation systems that often involve the patient, and the therapists [17].

There are researches that prove the sensorimotor control of limb stiffness and compliance is a key element in the organization of movement. Though, the best training approach would be the sensorimotor training. The goal of this therapy is to improve child's perceptual abilities intrinsic to the concurrent generation and experience of their own movements as well as use of this information to guide their movements [18].

This kind of training is difficult to apply with traditional therapeutic tools. However, various robot-assisted therapies are developed in order to support sensorimotor training, some of them including haptic devices. Robot-assisted therapies allowed to perform rehabilitation without the direct supervision of a trained therapist, what is especially important in DCD, where the number of patients significantly outgrows number of therapists. What is more, the complexity or/and difficulty of the task can be controlled very precisely.

Haptic devices were used to improve the generation of handwriting movements of children with DCD [19]. The researches show that this method can be notably effective, mostly because of active sensorimotor generation and control of movement trajectories. This is the feature required for learning generalized to task-related movements other than those specifically practiced [18]. This requirement is caused by catch-22 situation connected with movement learning mechanism. This learning process for a person without dyspraxia consists of two stages. First one is learning to generate imprecise approximation of movement. When this is achieved, one can gradually improve approximation of a movement through practice until achieving optimal route. For people with dyspraxia it is almost impossible to generate good enough qualitative approximation, which should allow them to make quantitative improvements through practice—this is a catch-22.

It is possible to overcome this paradox only if the support from the machine is balanced with possibility of active generation and control of movement from child. Mechanical properties of haptic devices, like inertia and tactile feedback can significantly help in achieving this result [18].

Using haptic devices as controllers in application allows to benefit from the various factors generated by particular devices. These factors vary between products. The Phantom Omni device consists of stylus connected to an arm mounted on base. It has six degrees of freedom and its workspace range is $160 \times 120 \times 70$ mm. Thanks to its pen-like stylus it is easy and natural to operate it, especially for children. Phantom Omni has been widely used in industry for a several years, also for medical purposes, thus it is a well-tested and safe-in-use device.

The haptic factors generated by this device and tested in children's rehabilitation are [18, 19]: inertia, viscosity (of the space), magnetic attraction (of the object), and friction (of the object). Viscosity and inertia may provide an appropriate support, but do not allow to keep track of designed movement form. Furthermore, no evidence suggests that the method could generalize the performance without support from the device [19]. Magnetic attraction help in following the generated path and this is the main haptic factor that will be used in the project. Researches show that friction has no significant impact on helping the child in performing the task [18].

4 Game Rules Analysis

The main task performed by patients is to follow the three-dimensional wire with tip of stylus, where the path is magnetically attractive. There is an object that the player has to move from start to end along the path by pushing it with the tip of stylus, represented by cursor. There is also a competitor's object, moving along the path with uniform velocity (specified for each curve). The player wins a round when his object reaches the end before competitor's one.

In order to develop a serious game which can help children with dyspraxia, some basic rules and presumption must be established. The following principles are based on [18]:

1. The game represents a bottom-up approach with sensorimotor training.
2. Basic task is to follow the path with tip of stylus, where the path is magnetically attractive.
3. The main haptic factor is magnetic attraction.
4. There must exist a levelling system.

5 Implementation

The game is implemented in C++ with OpenGL and Haptic Library API (under Academic License) to support Phantom Omni haptic device.

5.1 Path

The mathematical representation of path is the Bezier curve (Fig. 2). Depending on the level of difficulty, there are different curves—they vary in length, curvature, and torsion. There are three designed curves. Each curve has defined two times of passing—one for slower and one for faster race. The competitor's speed is calculated based on this time and the current path length.

Depending on the level of difficulty, curves have 7, 10, or 12 control points. The curve representation in the game is built from many single spheres, coordinates of canter are calculated in the following way:

$$(x, y, z) = \left(\sum_{i=0}^{n} x_i B_i^n(t), \sum_{i=0}^{n} y_i B_i^n(t), \sum_{i=0}^{n} z_i B_i^n(t) \right) \quad \text{for } t \in [0, 1] \tag{1}$$

where

$$B_i^n(t) = \begin{cases} \binom{n}{i} t^i (1-t)^{n-i} & \text{for } i = 0 \ldots n \\ 0 & \text{for } i < 0,\ i > n \end{cases} \tag{2}$$

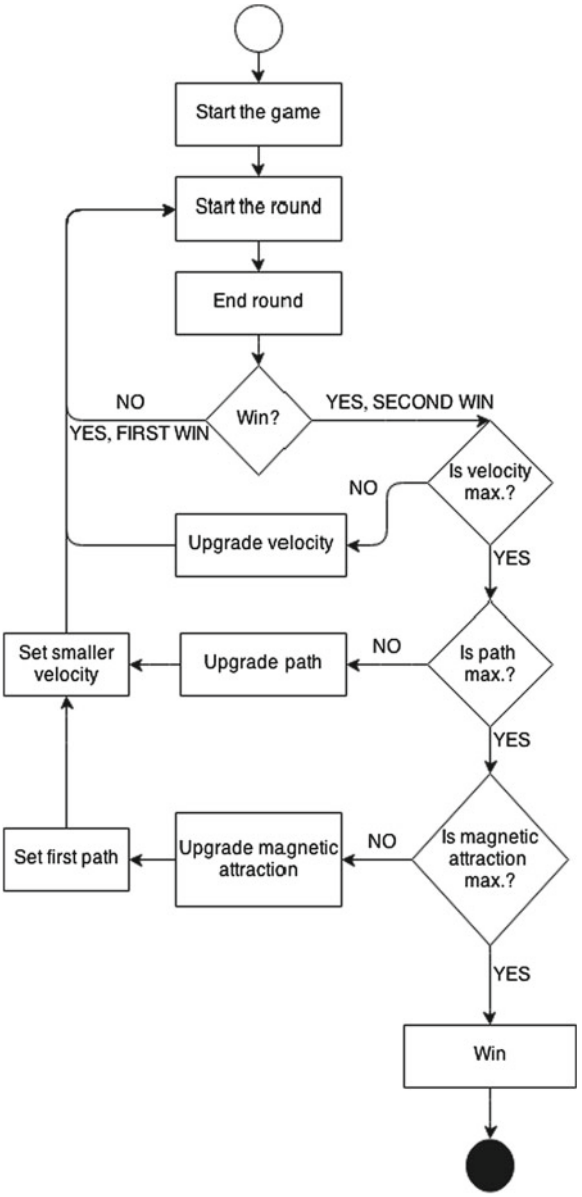


Fig. 1 Schema of leveling process

and $n - 1$—number of control points. The step of t incrementation is 1/500 = 0.002, because 500 spheres are generated for each curve.

5.2 Levelling up

There are three parameters that are changing along the game: the competitor's velocity, the path's shape (and length), the magnetic attraction level.

The player advances to the next level when he wins two times in a row with definite settings. While advancing, first, the competitor's velocity is increased. Then, when the player wins with both competitors, slow and fast one, he changes the path to more advance, but with slower competitor again. Finally, after winning games on all paths, the magnetic attraction level is decreased and player starts from the very first path with slower competitor. This process is repeated until reaching the lowest possible level of magnetic attraction and is presented in Fig. 1.

5.3 Measuring Player's Score

Continuous measuring player's score during a game is one of the most important parts of the project, as it allows to control his progress. There are two main values measured during the game: path length (of the object controlled by the player) and total race time.

The path length may differ from curve length in case player leaves the magnetic field of a wire. This affects more often children with dyspraxia. This parameter decreases while trainings and therefore is index of progress.

Total race time is a parameter responsible for determining of win. The counting starts from the moment when player's object moves for a first time in a round and lasts until reaching the end point.

5.4 Progress Detecting

Progress detecting is a process that repeats with 50 Hz frequency. It has to determine if the player is able to move his object or not by detecting the collision of path and cursor.

The information, that player is touching the path, comes from haptic thread. It has appropriate properties set if such event is happening. The problem is in checking on which part of path player is now. The whole path cannot be checked during the short time (not even 20 ms) because it would require checking collision with even 500 elements that the path consists of. The approximation is used, that in current moment of time application tests only 10 elements (points) that lie on the path just in front of the player's object—if the cursor collides with any of these points, object is moving to this point and the procedure is repeated until end of path.

The collision is calculated from sphere equation:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 <= r^2 \tag{3}$$

where x_0, y_0, z_0—current path element center coordinates; x, y, z—player coordinates; r—empirically determined radius with value 0.1.

5.5 Route Calculation

Player actual position and visualized position differ. Path magnetically attracts the cursor, so as a result there is a specified radius (different for each difficulty level) in which player can move without having his cursor unhook from the path. If player moves out of the range of the magnetic field, the cursor will move as well. Therefore, the route is measured based on the real movements, not just the ones displayed to player. The route is calculated using the following expression:

$$L = \sum_{i=1}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2} \tag{4}$$

where n—number of all points collected during measuring the route; x, y, z—point coordinates.

6 Game

In the Fig. 2 first path from game is presented. In Fig. 3 the player's object real route is showed. The sharp impulses are effects of sudden getting out of the range of magnetic attraction field (it requires pressure which is hard to control when the field disappears).

The game is ready to be presented to a children with dyspraxia as a rehabilitation tool. Although adding some more children-friendly features (like sounds, more colorful and friendly objects) may be considered to allows better interaction with patients, it is not required for application to work properly.

Fig. 2 First curve, the lowest level

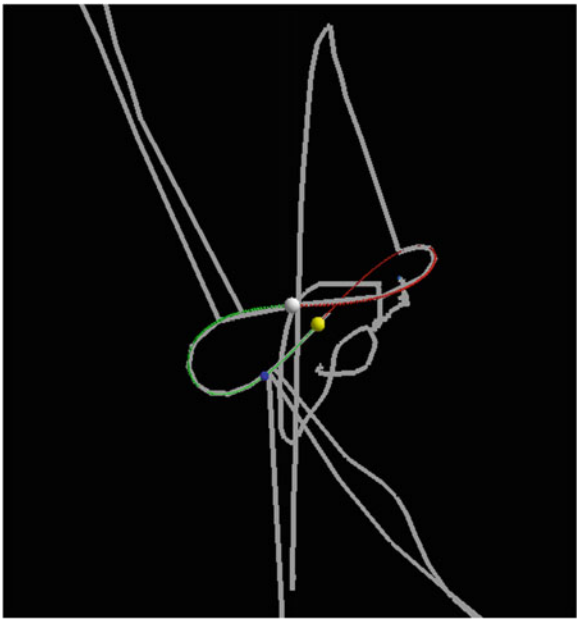


Fig. 3 The route of the player's object

7 Summary

The project was successful. All requirements have been implemented. Fully functional, stand-alone application has been created. The next stage of the project should be evaluation with children, so the game fulfills their requirements and then perform tests, in which application would be used as a rehabilitation tool for children with dyspraxia.

The Development Coordination Disorder is a troublesome disease, also because of psychological and social reasons. Engaging in the development of treatment for this disorder was a really valuable experience.

Acknowledgments This work was performed using the infrastructure supported by POIG.02.03.01-24-099/13 grant: "GCONiI—Upper-Silesian Center for Scientific Computation".

References

1. A. Henderson, N. Korner-Bitensky, M. Levin, Virtual reality in stroke rehabilitation: a systematic review of its effectiveness for upper limb motor recovery. Top. Stroke Rehabil. **14**(2), 52–61 (2007)
2. J. Broeren, K.S. Sunnerhagen, M. Rydmark, Haptic virtual rehabilitation in stroke: transferring research into clinical practice. Phys. Ther. Rev. **14**(5), 322 (2009)
3. J.G. Zwicker, C. Missiuna, S.R. Harris, L.A. Boyd, Developmental coordination disorder: a review and update. Eur. J. Paediatr. Neurol. **16**(6), 573–581 (2012)
4. M. Prensky, *Digital Game-Based Learning* (McGraw Hill, New York, 2001)
5. U. Ritterfeld, M. Cody, P. Vorderer, *Serious Games Mechanisms and Effects* (Routledge, London, 2009), pp. 486–512
6. D. Michael, S. Chen, *Serious Games: Games That Educate, Train, and Inform. Course Technology* (ACM, New York, 2005), pp. 1–313
7. P.M. Kato, Video games in health care: closing the gap. Rev. Gen. Psychol. **14**(2), 113–121 (2010)
8. A. Szczesna, Serious games in medicine. Bio-Algorithms Med-Syst. **9**(2), 97–102 (2013)
9. A. Szczesna, T. Grudzinski, J. Grudzinski, R. Mikuszewski, A. Debowski, The psychology serious game for pre-school children. *Proceedings of IEEE 1st International Conference on Serious Games and Applications for Health, SeGAH*, 57–60, 2011
10. A. Szczesna, M. Tomaszek, The methodology of designing serious games for children and adolescents focused on psychological goals, *Information Technologies in Biomedicine* (Springer, Berlin, 2012), pp. 245–255
11. T. Widiger, *American Psychiatric Association Diagnostic and Statistical Manual of Mental Disorders*, 4th edn. (American Psychiatric Association, Washington DC, 2000)
12. H.J. Polatajko, N. Cantin, Developmental coordination disorder (dyspraxia): an overview of the state of the art. Semin. Pediatri. Neurol. **12**(4), 250–258 (2005)
13. R.H. Geuze, M.J. Jongmans, M.M. Schoemaker, B. Smits-Engelsman, Clinical and research diagnostic criteria for developmental coordination disorder: a review and discussion. Hum. Mov. Sci. **20**(1), 7–47 (2001)
14. D. Dewey, B.N. Wilson, Developmental coordination disorder: What is it? Phys. Occup. Ther. Pediatr. **20**(2–3), 5–27 (2001)
15. N. Gordon, I. McKinlay, *Helping Clumsy Children* (Churchill Livingstone, Edinburgh, 1980)
16. D. Larkin, S.A. Cermak, Issues in identification and assessment of developmental coordination disorder, *Developmental Coordination Disorder* (Elsevier, Canada, 2002), pp. 86–102
17. M. Orozco, J. Silva, E. Petriu, A.E. Saddik, The role of haptics in games, in *Haptics Rendering and Applications*, ed. by A.E. Saddik (InTech, USA, 2012). ISBN 978-953-307-897-7
18. W. Snapp-Childs, M. Mon-Williams, G.P. Bingham, A sensorimotor approach to the training of manual actions in children with developmental coordination disorder. J. Child Neurol. **28**(2), 204–212 (2013)
19. H. Ben-Pazi, A. Ishihara, S. Kukke, T.D. Sanger, Increasing viscosity and inertia using a robotically controlled pen improves handwriting in children. J. Child Neurol. **25**(6), 674–680 (2010)

Parallel Biological In Silico Simulation

Patrick Amar, Muriel Baillieul, Dominique Barth, Bertrand LeCun, Franck Quessette and Sandrine Vial

Abstract It is crucial to understand how biological systems work, in particular the metabolic pathways, if we want to be able to understand diseases. One of the author has developed HSIM, a simulator dedicated to the biochemical simulation of the reactions inside the compartments of a virtual cell. Another author is the leader in the development of BOBPP a high level parallelization framework. In this article, we propose an important improvement of bobpp in order to run HSIM in parallel. This will allow us to simulate more complex models, involving more chemical species and reactions, leading to more realistic results using a smaller amount of computing time.

1 Introduction

The analysis of biological processes, such as metabolic pathways and signaling pathways are of great interest to understand the internal mechanisms of the cell. For instance, the metabolic and signaling pathways that lead to apoptosis (programmed death of the cell) are not very well understood for now, even if the different molecular compounds involved in apoptosis are well known. There are multiple apoptosis metabolic pathways, some of them seem to be quite independent from the

P. Amar (✉)
LRI, Paris-Sud University, Orsay cedex, France
e-mail: pa@lri.fr

M. Baillieul · D. Barth · B. LeCun · F. Quessette · S. Vial
PRiSM, Versailles University, Versailles, France
e-mail: murielbaillieul@prism.uvsq.fr

D. Barth
e-mail: dominiquebarth@prism.uvsq.fr

B. LeCun
e-mail: bertrandleCun@prism.uvsq.fr

F. Quessette
e-mail: franckquessette@prism.uvsq.fr

S. Vial
e-mail: sandrine.vial@prism.uvsq.fr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_40

others. Nevertheless, in some configurations, all these pathways are not correctly activated and the apoptosis does not occur. This is typically what happens in some cancers.

One of the promising approaches is the use of in silico simulations, since one may change the parameters and therefore be able to understand which are the key ones in intricate behaviors. There are two major difficulties: first, the number of acting molecules inside the cell (the simulation of a global cell is beyond our computation power for now) and second, the cell is naturally divided into organites, and each one may have a specific environment with its particularities.

HSIM is a cell simulator [1, 2] that allow us to simulate only the interesting actors of the cell without ignoring the effects of the others. In HSIM, organites with their membrane can be defined and taken into account. HSIM is detailed in Sect. 2. HSIM is written as a sequential simulation program, even if in a cell, distant molecules could be simulated in parallel.

BOBPP is a parallel library [3] that permits to simulate tasks in parallel. BOBPP is a high-level tool that makes programming independent of the computer that runs the resulting program. BOBPP was first designed for multi-core shared memory computers but the simulation program we wish to implement here leads us to make enhancements to BOBPP.

Section 2 gives a description of the two tools we used: HSIM and BOBPP. The cluster decomposition and the parallelization we make are described in Sect. 3. It is this decomposition, made according to both organelles and the division of space, that makes the originality of our contribution. Section 6 presents the modifications made to BOBPP in order to extend this platform to fit our approach.

2 Tools

In this section, we give a detailed description of both tools we used: HSIM (the simulation tool) and BOBPP (the parallel library).

2.1 HSIM

HSIM[1] is a simulation tool used to study the dynamics of biochemical processes in a virtual cell. The model is given using a language based on probabilistic rewriting rules that mimics the reactions between biochemical species. HSIM is a stochastic automaton which implements an entity-centered model of objects. This kind of modeling approach is an attractive alternative to differential equations for studying the diffusion and interactions of the many different proteins, enzymes, and metabolites in cells which may be present in either small or large number of copies. HSIM

[1] This software is freely available at http://www.lri.fr/~pa/Hsim.

has been developed since 2002 to study the assembly, the movements and the dissociation of large numbers of molecules in a virtual cell. The simulator is driven using a description of the model system written in a language that does not limit the simulation program to a particular model. For example, one can mix a metabolic network with a regulatory network and study the behavior of the whole model, including the interactions between the two kinds of networks. In essence, each molecule is represented by a record that includes its type, its position, its size, and a list of links to certain other molecules. HSIM keeps track of each molecule in real time from the computer point of view. The basic principle is that time is sliced into consecutive steps (or generations), and for each generation, the rules are applied to each molecule. These rules mimic the chemical reactions between molecules in a real organism. Metabolites diffuse faster than proteins because of their smaller size, this is taken into account in HSIM by representing them with a sphere of reduced size and with a greater diffusion speed. During a generation, the following processes are applied to all the molecules: the source molecule S is chosen at random (in order to avoid systematic artifacts); the diffusion step is made by picking a random direction (two angles in the 3D space) and by moving S, along this direction (random walk); If another molecule, T, intersects this line and if a reaction rule exists between a molecule of the type of S and a molecule of the type of T, this rule is applied, according to a probability representing the reaction kinetics; If not, molecule S is moved to the empty location, according to a probability representing its diffusion speed.

When all the molecules in the cell have been processed, this generation is completed, the current time is increased of the time slice, and a new generation may begin. In HSIM the computer time is only proportional to the total number of molecules (and not to the size of the simulated space or the number of types of molecules or the number of reactions).

The model is described in a configuration file that contains the size of the compartment, the size and location of the subcompartments, the declaration of each kind of molecules (the name of the species, the global maximum number of links to any kind of molecules), if it is a membrane molecule or a cytosolic molecule, its diffusion speed, its size, the maximum number of potential links to each specific kind of molecules, the reaction rules and the initial population of each kind of molecules.

2.2 BOBPP

This section presents BOBPP, a parallel high-level frameworks that facilitate the development of parallel applications. Most of time frameworks are written for one or another fixed structure of tasks. Frameworks like Kaapi [4] or Charm++ [5] execute

Data Flow Graph (DFG). In this case, the application is decomposed in elementary tasks (nodes of graph). The arcs symbolize the data dependencies between tasks. Tasks can be dynamically created, but when a task has been executed one time, it is deleted from the set of tasks. Many other frameworks also exist but are developed to execute tree of tasks. BOBPP [3, 6, 7], one of these, has been developed to facilitate the development of parallel code to solve problems belonging to the combinatorial optimization domain. These problems belong to the $\mathcal{NP}$-*Hard* complexity class of problems, they require an exponential solving time in the worst case. Then, after having designed the best sequential algorithm to solve a problem, according to the existing computers, it is natural to reduce the computation time using a parallel machine.

Several software frameworks have been proposed for solving Combinatorial Optimization problems. Like BOBPP [3], they also establish the interface between the user code and the parallel machines. These tools include BCP [8], PEBBL [9] ALPS [10, 11], Bob [12, 13], and others. But all of them only deals with Branch and Bound like algorithms. Furthermore, they only propose one parallelization, which is suitable for one type of machine and generally one kind of application.

BOBPP has been developed as an interface between the user code and the parallel machine. It has proposed different flexible tree search algorithms (Branch and X, Divide and Conquer) and also different parallelizations based on PTHREADS, MPI. From the parallelization point of view, BOBPP can be viewed as a framework where it is possible to create or delete tasks named Nodes that are scheduled dynamically on the computing cores or threads. Originally, the aim of the execution of a BOBPP Node is to create new node that will be scheduled later. All pending nodes are stored in a Global Priority Queue (GPQ), where the threads pick up their work. In other terms, this GPQ represents the global scheduler of the tasks. Once a task is executed it is deleted. As we will see in the Sect. 3, the biological simulation in HSIM needs to keep the tasks alive.

Then the performance and the suitability of the application depends on the algorithm used to manage the GPQ. In shared memory machine (SMP), with few cores, the GPQ may be a simple priority queue (PQ) protected by mutual exclusion primitives. In bigger SMP, to breakout the bottleneck using too many threads on the same PQ, the GPQ may be composed by several PQ. Threads are partitioned in several groups. Each group is assigned to one PQ (as shown in the Fig. 1). On a distributed memory machine (DSM), each process may have several threads. These threads may access to one or several PQ. In such a case, an explicit load balancing is done between PQ. The best choice for the number of PQs during the solving depends on the nature of the problem: some problems develop big tasks (in term of computation time or space required to store them) and other problems develop "small" tasks (in term of space in memory or computation time).

In BOBPP, the SMP implementation is based on POSIX threads, although on DSM the implementation is based on both POSIX-threads and MPI. From the interface point of view, the user chooses the parallel environment (pthreads or MPI+pthreads) he wants by calling the appropriate BOBPP classes or methods in his code.

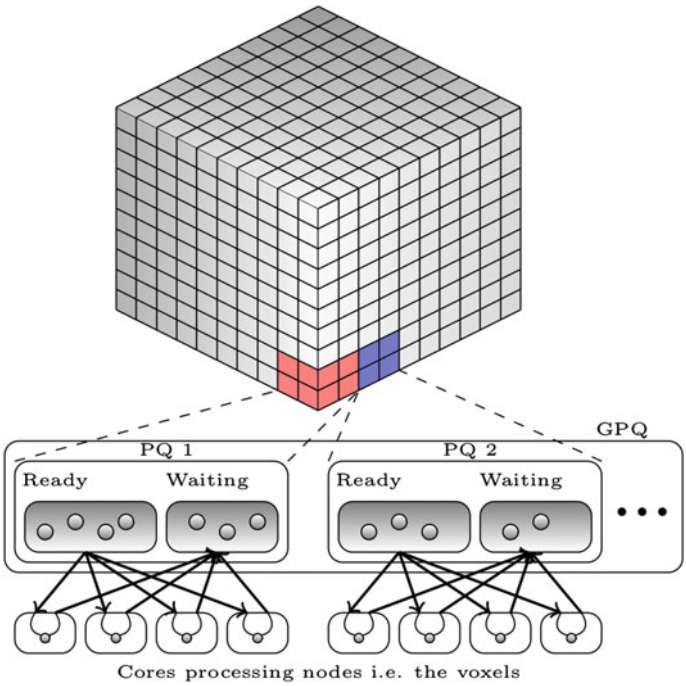


Fig. 1 Porting HSIM on *top* of BOBPP

3 The Voxel Decomposition

The decomposition of the tasks to execute is not made according to a load equilibrium but takes into account the topology of the cell. The inputs of the program are:

- Dimensions of the cell, thickness of the cell membrane.
- The entities present in the cell, for each one is given: id, name, dimensions, distance of diffusion per iteration.
- The HSIM rules of reactions.
- The organelles: this is the fundamental part since one can separate a cell in functional parts (compartments) such as nucleus, mitochondria, etc. For each compartment, the dimensions, the thickness of its membrane, and the position in the cell are given. The organelles will induce the load balancing and a set of tasks will be dedicated to each organelle but no tasks will have to deal with different organelles. Furthermore, some entities are present only in some organelles, thus we create a subprogram for the entities and the HSIM rules concerning only the entities present in the organelle. In some parts of some membranes, the diffusion of membrane protein is made only in two dimensions.

- The quantity/concentration/occupation rate of each entity in each organelle and the set of organelles it may be present. It is by default restricted to the organelles where the entity is present at the initialization.
- The voxel grid that split the cell. Note that a voxel cannot cross over two organelles. We first split the cell in organelle and then the organelle are split according to the voxels. Each voxel also knows the identities of the six voxels in its neighborhood.
- The number of iterations.
- The log frequency of the statistics.

Each voxel is assigned to a BOBPP task Fig. 1. At each iteration, when a task is ready, the task first retrieve the entities sent by its neighbors. Then the task applies the rules of diffusions and reactions to all the entities inside its voxel. At the end, some entities may move to a neighbor voxel. These entities and the date of voxel change, are stored in the task assigned to the corresponding neighbor's voxel. Once the task has finished its iteration, it becomes a waiting task. A task becomes ready again and can execute its nth step only if its neighbors have terminated their $(n-1)$th step. Thus, we do not need to have a global synchronization, since the synchronization is automatically made as follows. When a task has finished its th step, it informs its six neighbors. When a task has received this information from all its neighbors it becomes ready.

These communications between tasks and the activation process were not possible in the current version of BOBPP. The next section explains the implementation of these new features in BOBPP.

4 Bob++ Enhancement

As explained in Sect. 2.2, BOBPP has not been written to handle parallel application like HSIM.

BOBPP has been extended for this application. Due to the BOBPP design, we have extended the BOBPP framework by adding new algorithm to the framework called SimSp for simulation space. The SimpSp algorithm consists in picking up one ready node executing it and reinserting it in the GPQ. In this implementation, all nodes are initially created to cover the simulation space. Each node is inserted in the GPQ.

A specific bi-directional communication canal called a `Link` must be declared between two nodes that exchange particles. For a two-dimensional space, each node must have four links, for a 3-D space, each node has six links. A node is ready for the iteration n if it has message from all its links at the iteration $n-1$ At the beginning of an iteration, a node gets all the messages from its `Links`. And at the end of its iteration, it sends messages on all its links. Then as shown in the Fig. 1, we have also added a specific implementation of priority queue which is composed by two queues to handle the two node's states: waiting and ready. The first one the waiting queue stores the waiting nodes, although the second one called the ready queue stores the ready nodes. When a node become ready, it is deleted from the waiting queue and is

inserted in the ready queue. This implementation is asynchronous, the mechanism of `Link` with tagged messages ensures that a node has all the necessary information to complete the execution at the iteration n. Then the synchronicity is only local, between pair of nodes. There is no need of time-consuming mechanism like global barrier or global mechanism to be sure that the execution is correct. It is easy to see that this local synchronous mechanism ensures a correct global execution of the system.

The partitioning of the global space in more parts than the number of cores, permits a good load balancing. In our implementation, we have not fixed that the number of nodes, i.e., the number of tasks must be equal to the number of computing cores. As described in the Sect. 2.2, a fixed number of cores is associated to a a PQ. But the PQ can store many more nodes that the number of computing cores. Then a core has work while a ready node exists in the PQ. This is classic way to balance the load between cores. While one core executes a long task, another core can execute several small tasks. That reduces the idle time of cores waiting that a long task is finished.

5 Example of a Simulation

We model a cell of $1\,\mu\text{m} \times 1\,\mu\text{m} \times 1\,\mu\text{m}$ with a membrane of 7.5 nm of thickness. In this cell, we use only one organelle (nucleus) of $100\,\text{nm} \times 100\,\text{nm} \times 100\,\text{nm}$ with a membrane of 7.5 nm of thickness. We use a 3D-grid of 1000 voxels. Entities are the following with their diameter and diffusion time per iteration: $E1$ (7 nm, 7 nm/it.), $S1$ (10 nm, 6 nm/it.), $E1S1$ (17 nm, 7 nm/it.), $E1P1$ (17 nm, 8 nm/it.), $P1$ (10 nm, 6.5 nm/it.). There are three possible reactions : $E1 + S1 \rightarrow E1S1$ with probability 0.7; $E1S1 \rightarrow E1P1$ with probability 0.9; and $E1P1 \rightarrow E1 + P1$ with probability 0.8. At the beginning, in the nucleus there is 30 % of volume occupied by $E1$ and 50 % occupied by $S1$. In the rest of the cell, $E1$ occupied 20 % of volume and S1, 50 % of volume. We simulate five iterations by voxels. At the end, there was 65,752 entities that change of voxels and 8,34,106 collisions. At the beginning, there was 1,41,790 $E1$ and 1,21,506 $S1$ and no other entities. At the end of the simulation, there was 97,825 $E1$, 77,539 $S1$, 43,946 $E1S1$, 19 $E1P1$, and 2 $P1$. This example of a very simple example shows us that our approach is promising. We need now to validate our approach by a biological expertise.

6 Conclusion and Perspectives

We propose a generic parallel tool for in silico simulation of complex biological processes in complex environment (organelles). To be able to run HSIM above BOBPP gives access to automatic and efficient parallelization with the powerful simulation description of HSIM.

We would like to port the BOBPP SimSp algorithm on the BOBPP distributed environment. At this time, the link communication is not mapped on MPI communication link. According to the machine architecture, the management of messages can be different. On machine with small network latency, a MPI message maybe sent for each Link message. But on machines with high network latency, the crucial point is be able to group the BOBPP Link messages into one MPI message to reduce the overhead.

The first results are very encouraging and biological expertise has already confirmed the relevance of HSIM. What is clearly needed is the possibility to simulate large systems. As future work, we will launch large tests on parallel computers. The use of GPU is also studied since in a voxel, a lot of entities have to run the same HSIM rules in parallel.

Acknowledgments This work was partially supported by grant ANR MARMOTE (ANR-12-MONU-0019).

References

1. P. Amar, G. Bernot, V. Norris, Hsim: a simualtion programme to study large assemblies of proteins. J. Biol. Phys. Chem. **4**(2), 50–63 (2004)
2. P. Amar, L. Paulevé, Hsim: an hybrid stochastic simulation system for systems biology. in: *the Third International Conference on Static Analysis and Systems Biology* (2012)
3. F. Galea, B. Le Cun, Bob++ : a framework for exact combinatorial optimization methods on parallel machines. in: *International Conference High Performance Computing & Simulation 2007 (HPCS'07)*, pp. 779–785 (2007)
4. T. Gautier, J.V.F. Lima, N. Maillard, B. Raffin, XKaapi: a Runtime System for Data-Flow Task Programming on Heterogeneous Architectures. in: *Proc. of the 27-th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. Boston, USA (2013)
5. L. Kale, B. Ramkunar, A. Sinha, A. Gursoy, The charm parallel programming language and system. Tech. Rep. **2**, 1–27 (1994)
6. T. Crainic, B. Le Cun, C. Roucairol, *Parallel Branch and Bound Algorithms* (Wiley, USA, 2006), pp. 1–28
7. T. Menouer, B. Le Cun, P. Vander-Swalmen, Solving combinatorial problems on hpc with bobpp, WolfHPC in SuperComputing (2012)
8. M.J. Saltzman, Coin-or, *Programming Languages and Systems in Computational Economics and Finance*, An Open Source Library for Optimization (Springer, Boston, 2002)
9. J. Eckstein, C.A. Phillips, W.E. Hart, PEBBL 1.0 User Guide. RRR 19–2006, RUTCOR (2006)
10. T. Ralphs, L. Ladányi, M. Saltzman, A Library Hierarchy for Implementing Scalable Parallel Search algorithms. J. Supercomp. **28**(2), 215–234 (2004)
11. Y. Xu, T. Ralphs, L. Ladányi, M. Saltzman, ALPS: A Framework for Implementing Parallel Search Algorithms. in: *Proc. of the Ninth INFORMS Computing Society Conference* (2005)
12. V.D. Cung, S. Dowaji, B. Le C, T. Mautor, C. Roucairol, Concurrent data structures and load balancing strategies for parallel branch-and-bound/a* algorithms. in: *III Annual Implementation Challenge Workshop, DIMACS*. New Brunswick, USA (1994)
13. B. Le Cun, C. Roucairol, the PNN team: Bob : a unified platform for implementing branch-and-bound like algorithms. RR 95/16, Laboratoire PR iSM, Université de Versailles - Saint Quentin en Yvelines (1995)

An Analysis on Empirical Performance of SSD-Based RAID

Chanhyun Park, Seongjin Lee and Youjip Won

Abstract In this paper, we measure the I/O performance of five filesystems—EXT4, XFS, BTRFS, NILFS2, and F2FS, with five storage configurations—single SSD, RAID 0, RAID 1, RAID 10, and RAID 5. We show that F2FS on RAID 0 and RAID 5 with eight SSDs outperforms EXT4 by 5 times and 50 times, respectively. We also make a case that RAID controller can be a significant bottleneck in building a RAID system with high speed SSDs.

Keywords RAID · SSD · Disk I/O performance · Filesystem · F2FS

1 Introduction

A Solid State Drive (SSD) is a low power storage device with high I/O bandwidth that has received much attention as a device that may replace Hard Disk Drives (HDDs) and remove I/O performance bottleneck of a computer [1–5]. As the use of services that require reliability and rapid response expands, the demand for a device that meets stringent I/O performance requirements are also increasing [6, 7]. RAID (Redundant Array of Inexpensive Disks) [8] exploits slow storage device that is HDD to improve the I/O performance of a system. One of its strength is its customizability based on the required level of reliability and performance of a computing system.

Recent studies on RAID try to use SSD as alternative to HDD and explore performance of various RAID configuration using the SSD, such as effect of stripe size

C. Park (✉) · S. Lee · Y. Won
Department of Computer and Software, Hanyang University, Seoul, Korea
e-mail: parkch0708@hanyang.ac.kr

S. Lee
e-mail: insight@hanyang.ac.kr

Y. Won
e-mail: yjwon@hanyang.ac.kr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_41

[9–11] or RAID level [3, 7, 11]. However, to the best of our knowledge, the research community has not thoroughly analyzed the effect of software such as filesystem layer in the I/O hierarchy for SSD-based RAID storage system. In this research, we establish a hypothesis that log-structured filesystem is more suitable than journaling filesystem or copy-on-write filesystem for SSD-based RAID. To prove the hypothesis, we measure the I/O performance of two journaling filesystems (EXT4 [12], XFS [13]), one copy-on-write filesystem (BTRFS [14]) and two log-structured filesystems (NILFS2 [15], F2FS [16]) on SSD-based RAID. In order to provide fair experiment conditions for each filesystems, we first obtain the optimal stripe size. After obtaining the optimal stripe size, RAID organization, and the number of disks from the experiment, we analyzed I/O performances on five filesystems (EXT4, XFS, BTRFS, NILFS2, and F2FS). This experiment showed that the selection of filesystem can draw a difference in performance for RAID organization by more than 50 times.

2 Related Work

In relation to the organization of RAID with existing SSDs, many studies have been conducted [1–3, 7, 9–11].

Some studies showed the impact of stripe size on the I/O performance of SSD-based RAID [9–11]. These researches showed a correlation between the stripe size and the I/O performance. One of these study [11] in which I/O performances were measured under various stripe sizes, 16 KB, 64 KB, and 1,024 KB, with record sizes from 8 KB to 256 KB. While changing stripe size, sequential read performance showed differences of more than 250 MB/s; sequential write performance showed differences of more than 300 MB/s; random read performance showed differences of more than 7000IOPS; and random write performance showed differences of more than 2000IOPS in a particular record size.

And there are some studies that analyze the changes in I/O performance when RAID levels and the number of SSDs organizing RAID are changed [3, 7, 11]. Among them, some studies confirmed that RAID organization with SSDs can make effects different from HDD on performance of RAID 5 [3]. During the write work, read work is added to RAID 5 due to the characteristics. In this research, when organizing RAID 5 with HDD, whose read performance and write performance are symmetrical, the performance is reduced more greatly than RAID 5 such as RAID0, RAID10, etc., because of the characteristics. On the contrary, if RAID 5 consists of SSDs, which performs better at reading than at writing, it shows no performance difference from other RAID levels, such as RAID 0 or RAID 10.

For the methods embodying RAID, there is a software RAID as an operating system which organizes and manages RAID, and a hardware RAID as separate equipment, RAID controller, which organizes and manages RAID. Among previous research, there were some studies which organized software RAID and measured the performance [7, 9]. One research [7] mentioned that maximum limitation of I/O

performance can be caused by bottleneck of RAID controller during the organization of hardware RAID with SSDs, and suggested the possibility of incongruity of hardware RAID organized by SSDs.

Much research has been conducted on SSD-based RAID but there was no proper research on the effects of software class on RAID organization. We considered filesystem one of the factors that must be considered during RAID organization.

3 Background

3.1 RAID

RAID combines several disks into one logical storage so as to use it as one disk with large capacity [8]. RAID divides the requested I/Os into certain size called stripe unit [17] or chunk, and distributes them in multiple disks. It is very important to select the optimal stripe size because stripe size cannot be changed dynamically [7].

RAID 0 distributes the data to number of disks used to organize the RAID system. If a large sequential I/O is issued, the RAID controller segments the I/O to a stripe size and writes them to disks in the RAID system in parallel. The process of dividing the data and distributing the stripes to number of disks is called striping. RAID 0 is the fastest RAID system because it maximizes parallelism, and it also affords the largest capacity among RAID systems; however, the integrity of the system breaks if a disk in the RAID 0 fails.

RAID 1 is often called as mirroring which requires two disks. Identical data is stored on each disk. Since same data is stored on mirrored disk, the system can withstand the failure in any of the disk. But, benefit of using RAID 1 comes in great cost, which limits the user space to 50 % of total storage capacity.

RAID 10 creates mirror of stripe sets that is applying RAID 0 on RAID 1; to create RAID 10 system, at least four disks are required—two for striping and the other two for mirroring of the stripe set.

RAID 5 exploits striping with distributed parity. In order to configure the system, it requires at least three disks, two for the stripe unit and one for the parity. Since it keeps a parity, the system can be recovered from failure in any one of the disk; however, the total storage space available reduces to store the parity.

3.2 Filesystem Synopsis

Journaling filesystem, such as EXT4 [12] or XFS [13], saves information of all write requests in the journal area with a particular form. There are three journal modes in journaling filesystem: journal, ordered, and write back mode. Journal mode saves both data and metadata in the journal area; ordered mode records changes in metadata

to the journal area only after data is written to the storage. Write back mode writes metadata in the journal area and keeps data on the main filesystem; however, write back mode disregards the ordering of the data. Note that journal area is placed in the middle LBA of a partition to minimize the distance of movement of HDD's arm. When the arm moves back and forth to record journal data and filesystem data, the locality of the filesystem data can be broken. One way to avoid the break of the locality is to use external journaling which exploits a separate disk as a medium for storing the journal data. This research uses ordered mode to examine effect of journal on RAID system.

BTRFS [14] is a copy-on-write (COW) filesystem which is introduced in Linux Kernel 2.6.29. It is also known as next generation filesystem that provides features such as built-in volume management, per-block checksumming, self-healing redundant arrays, and atomic copy-on-write snapshots. From the perspective of I/O count and I/O volume, copy-on-write feature can be an issue. In other filesystems that does not use copy-on-write, the filesystem overwrites the existing data on the storage, whereas the copy-on-write filesystem does not overwrite the existing data, instead it writes the new data to elsewhere. This research intends to examine the overhead of managing metadata and write requests when copy-on-write filesystem is applied to RAID and repeated updates must be treated.

NILFS2 [15] and F2FS [16] are log-structured filesystem which is merged to Linux mainline in Kernel 2.6.30 and Kernel 3.8, respectively. Log-structured filesystem appends all incoming data to the end of the log, which is in units of segment. Since all write requests are treated as sequential write operations, bandwidth of random writes on log-structured filesystem exhibits the same performance as sequential write; however, a critical problem with the log-structured filesystem is its read performance. Since all writes are written sequentially regardless of its spatial locality, all data must be read randomly. This research intends to examine the performance benefits of log-structured filesystem in SSD-based RAID.

4 Environment

The objective of this study is to analyze the behavior of SSD-based RAID under various RAID configurations, and examine the effect of stripe size and filesystem on storage performance. In this paper, we use five stripe sizes—64, 128, 256, 512, and 1,024 KB, and use five storage configuration—single SSD, RAID 0, RAID 1, RAID 10, and RAID 5. Workloads used in this paper are sequential read/write and random read/write, which are tested with buffered I/O and direct I/O.

We use MobiBench [18] to measure the I/O performance of the RAID system. We use 5 % of available filesystem partition as a file size throughout the experiment, and the I/O size for sequential and random I/O is set to 2 MB and 4 KB, respectively.

To measure the performance of the system, we used a computer that consists of eight 256 GB SATA3.0 Samsung SSD 840 Pro, connected to a Dell PERC H710P (1GB Cache memory, 2 SAS ports) on PCI-E 2.0 8 lane interface in a system with

16GB of memory (Samsung DDR3 SDRAM PC3 12800 4 GB × 4) and a Intel Core i7-3770 (4 cores with clock speed of 3.5 GHz). The system operates on Ubuntu 13.04 64 bit, Kernel 3.13. The maximum performance of a SSD 840 Pro 256 GB for sequential read/write is 540/520 MB/s and for random read/write is 100K/90KIOPS.

Theoretically, the maximum bandwidth of write operation on RAID 0 with N number of disks can be calculated as the number of disks times bandwidth of a device. Suppose a system exploits a SSD that has read and write bandwidth of 540 MB/s and 520 MB/s, respectively, the performance of a RAID 0 with 8 SSDs is 4,320 MB/s and 4,160 MB/s, respectively. The maximum bandwidth of H710P connected to PCI-E 2.0 × 8 lane can be calculated as bandwidth of PCI-E 2.0, 6 Gbit/s times 8 lanes, which is 4,096 MB/s. From the fact that the maximum interface bandwidth is lower than the maximum bandwidth of RAID 0 with 8 SSDs, we can deduce that the interface can be a source of the bottleneck on a RAID system with very high bandwidth.

5 Experiment

5.1 Effect of Stripe Size

We configure RAID 0 and RAID 5 with eight Samsung SSD 840 Pro, and vary the stripe unit size to examine the I/O performance. We measure sequential I/O performance (MB/s) as well as random I/O performance (KIOPS).

Figure 1 illustrates the result of I/O bandwidth of different stripe sizes ranging from 64 KB to 1,024 KB in multiples of two for RAID 0 and RAID 5 with eight SSDs. The performance of RAID 0 is shown in Fig. 1a, b. The best sequential I/O performance in RAID 0 is observed when the stripe size is 512 KB in both read and write, except for the case of sequential buffered read. For sequential read, buffered I/O and direct I/O yield 1,585 MB/s and 2,149 MB/s, respectively; on the other hand, sequential write operation with buffered I/O and direct I/O yield 1,774 MB/s and 2,535 MB/s, respectively. We observe that the memory copy overhead of buffered I/O brings about 25–30 % performance degradation in sequential read and write, respectively. In the case of random read, buffered I/O and direct I/O yield 8.3KIOPS and 8.4KIOPS, respectively. And, random write with buffered I/O and direct I/O yield 69.3KIOPS and 21.2KIOPS, respectively.

Performance on RAID 5 is shown in Fig. 1c, d. We observe that stripe size of 512 KB also shows best performance on RAID 5, except for random buffered write. For the sequential read with buffered I/O and direct I/O yield 1,633 MB/s and 2,089 MB/s, respectively. And, the sequential write with buffered I/O and direct I/O yield 1,859 MB/s and 2,097 MB/s, respectively. For the random read, on the other hand, buffered I/O and direct I/O yield 8.2KIOPS and 8.4KIOPS, respectively. For the random write, buffered I/O and direct I/O yield 3.9KIOPS and 10.8KIOPS, respectively. Stripe size of 64 KB on random buffered write, which exhibits the best performance in the test, shows throughput of 15.3KIOPS.

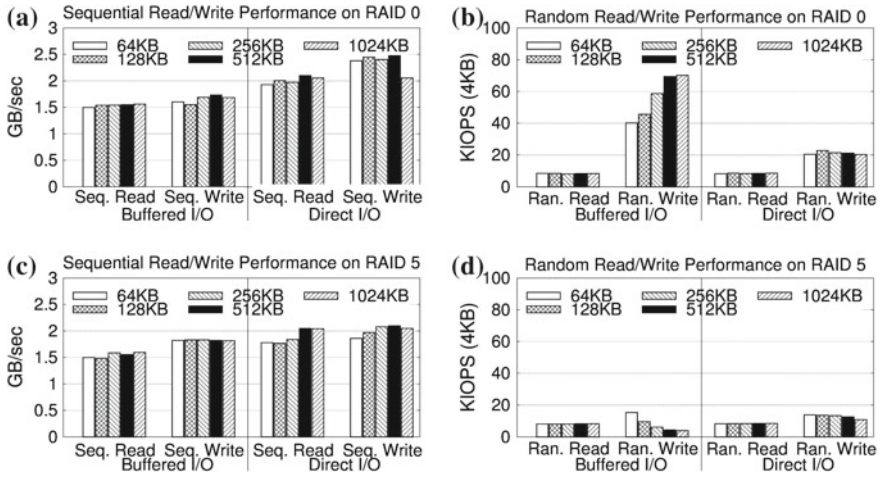


Fig. 1 **a** Seq. read/write performance on RAID 0. **b** Ran. read/write performance on RAID 0. **c** Seq. read/write performance on RAID 5. **d** Ran. read/write performance on RAID 5. I/O performance by stripe size on RAID 0 and RAID 5 (SSDx8, EXT4 filesystem)

We observe that stripe size of 512 KB exhibits either the best performance or equivalently good performance compared to the best case. Therefore, we set the optimal stripe size is 512 KB for sequential read/write and random read/write on both buffered I/O and direct I/O, except for random buffered write. The next set of experiments use stripe size of 512 KB.

5.2 Effect of RAID Level

Figure 2 illustrates the I/O performance with respect to RAID levels and the number of SSDs. Although it is not shown in the graph, we also measured the performance of RAID 1 and RAID 10. The performance of RAID 0 with one SSD and RAID 1 with two SSDs shows similar performance to sequential read/write performance with single SSD. Bandwidth of sequential write shows about 10 % lower performance than the performance of single SSD. RAID organization with additional number of disks shows better I/O performances than that of single SSD.

As the number of SSDs in RAID configuration increases, the I/O performance improves to a certain level. However, once the peak is reached, additional increase in the number of SSDs does not bring better performance. For sequential buffered read on RAID 0, the performance reached about 1,600 MB/s with three SSDs and stayed at that level even with more SSDs. In the case of sequential direct read, about 2,100 MB/s was reached with five SSDs. For sequential write on RAID 0 with four SSDs, buffered I/O shows about 1,800 MB/s and direct I/O with six SSDs shows about 2,500 MB/s. The result of RAID 0 experiment shows that the maximum

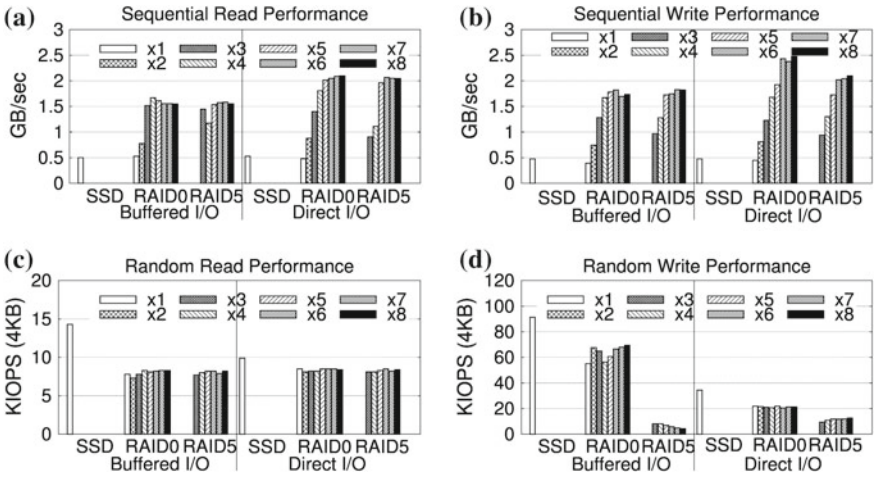


Fig. 2 **a** Sequential read. **b** Sequential write. **c** Random read. **d** Random write. I/O performance by RAID level and the number of SSDs (EXT4 filesystem)

performance of sequential read and write of RAID controller is about 2,100 MB/s and 2,500 MB/s, respectively. It is interesting to see that the bandwidth of sequential workload on RAID 5 is not far different from the performance of RAID 0. The performance result of changing the number of disks in RAID 0 and RAID 5 implies that the performance bottleneck lies in the RAID controller.

Random read and write workload on both RAID 0 and RAID 5 shows inferior performance compared to that of single SSD. In the case of random read performance, the number of disks in RAID configuration does not affect the performance. With single SSD, random read with buffered I/O and direct I/O shows 14.3KIOPS and 9.9KIOPS, respectively. We observe that the performance of RAID 0 and RAID 5 is almost the same regardless of the number of SSDs used; random buffered read exhibits minimum of 7.8KIOPS and maximum of 8.7KIOPS, and random direct read shows minimum of 8.1KIOPS and maximum of 8.5KIOPS. On all RAID configurations, random buffered read shows about 40–45 % lower performance compared to the performance of single SSD, and random direct read shows about 14–18 % lower performance to that of single SSD.

The performance of random write shows severe performance reduction in RAID 5. For random buffered write, RAID 0 shows about 20–40 % lower performance compared to single SSD; RAID 1 and RAID 10 shows performance reduction of about 55 %. The I/O performance of RAID 5 is reduced about 90–95 % compared to single SSD. In the case of random direct write on RAID 0, RAID 1, and RAID 10, about 38 % of I/O performance is reduced compared to that of single SSD; in the contrary, about 67 % of performance is reduced in RAID 5.

There are two interesting findings we can deduce from the result of experiment on the I/O performance with respect to RAID level and the number of SSDs in

RAID configuration. First, measured I/O performance does not match the theoretical performance measurements calculated with respect to RAID levels. In fact, our measurements show that the maximum I/O performance is lower than the theoretical measurements. We believe that the RAID controller is the main bottleneck in limited I/O performance. Second, we find that random read/write performance of RAID is much lower than that of single SSD—random buffered write on RAID 5 is about 95 % lower than the performance of single SSD. We believe random performance is slow because RAID cannot exploit parallelism while process the random I/O requests.

5.3 *Effect of Filesystem*

Figure 3 shows I/O performance of different filesystems on RAID 0 with eight SSDs. In the case of sequential read performance, both buffered I/O and direct I/O shows I/O performance of about 500 MB/s in all of five filesystems. It is interesting to see that sequential buffered read on BTRFS shows the best performance of about 2,500 MB/s, which is 160 % more than that of other filesystems on RAID 0. It shows that the performance of RAID 0 is about three times higher than that of single SSD. With direct I/O, the performance of RAID 0 is higher than that of single SSD by about 4.2 times in all filesystems except for BTRFS. The four filesystems, except BTRFS, show I/O performance of about 2,100 MB/s; BTRFS exhibits performance of about 1,700 MB/s.

Performance of sequential buffered write measured on single SSD shows I/O performance of about 500 MB/s in four filesystems except for NILFS2. In the case of

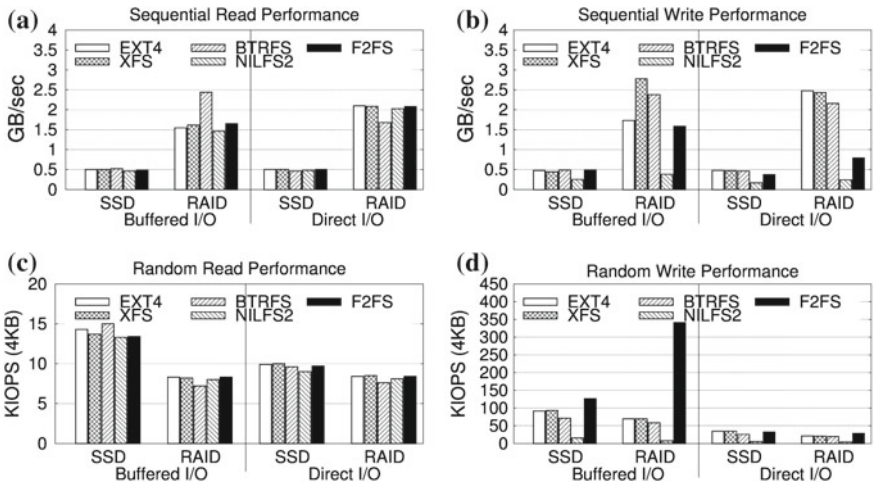


Fig. 3 **a** Sequential read. **b** Sequential write. **c** Random read. **d** Random write. I/O performance by filesystem (RAID 0, SSDx8)

direct I/O, all filesystems other than F2FS shows performance of 500 MB/s; performance of F2FS is about 380 MB/s. In the case of RAID, XFS has highest I/O performance of about 2,850 MB/s in buffered I/O, and EXT4 and XFS is the filesystem with highest I/O performance, producing performance of about 2,500 MB/s in direct I/O. BTRFS shows the second highest performance in both buffered I/O and direct I/O with 2,400 MB/s and 2,200 MB/s, respectively. In the case of F2FS, the performance of buffered I/O on RAID 0 is about 3.2 times better than that of single SSD, but the performance of direct I/O is 2.1 time better than that of single SSD.

For random read performance, the extent of I/O performance reduction is similar in all filesystems in comparison with single SSD. With RAID 0, random buffered read performance decreases by about 40 % compared to that of single SSD, and random direct read shows about 15 % drop in performance. In the case of random direct write, the performance of RAID is measured at about 60 % of single SSD in all filesystems except for NILFS2.

The performance of random buffered write shows the most interesting result. It shows that only F2FS on RAID 0 exceeds the performance of single SSD, whereas the performance of the other filesystems are lower than that of single SSD. The performance of the four other filesystems show about 20–50 % lower performance than the performance measured on single SSD. On the contrary, F2FS on RAID 0 shows 2.7 times better I/O performance than single SSD.

Figure 4 compares the I/O performance between EXT4 and F2FS in single SSD, RAID 0 (SSDx8) and RAID 5 (SSDx8). The result shows striking difference of performance on sequential direct write and random buffered write. In the case of sequential direct write, the performance of F2FS is lower than that of EXT4 on SSD, RAID 0, and RAID 5. Although sequential write with direct I/O of EXT4 on RAID 5 is about 20 % lower than that of RAID 0, the performance on RAID 0 and RAID 5 is about 3.1 and 2.6 times better than that of F2FS, respectively.

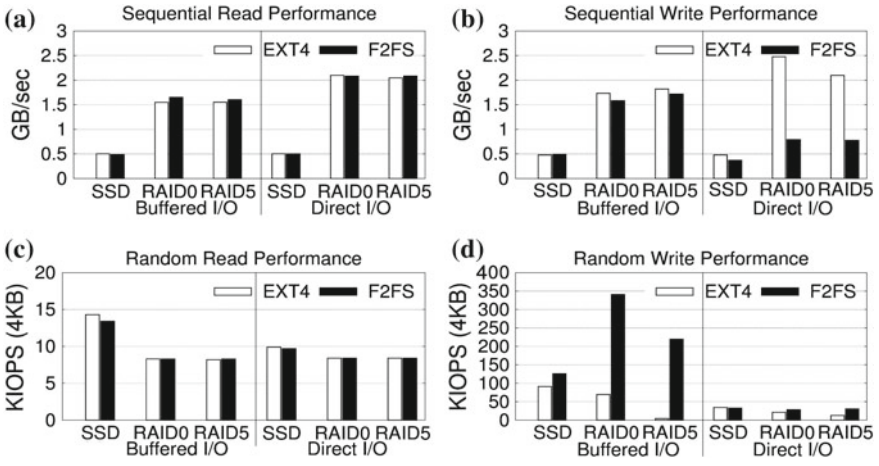


Fig. 4 **a** Sequential read. **b** Sequential write. **c** Random read. **d** Random write. I/O performance on EXT4 and F2FS on RAID 0 (SSDx8) and RAID 5 (SSDx8)

We observe that the performance of random write with buffered I/O on EXT4 does not excel as much as in the sequential write experiment. In fact, the performance of F2FS on RAID 0 and RAID 5 is about 5 times and 50 times better than that of EXT4, respectively. It is interesting to see that the performance of EXT4 on RAID 5 is very poor; the throughput of EXT4 on random write with buffered I/O shows about 40 times lower compared to the performance on single SSD. Although F2FS shows lower I/O performance on RAID 5 compared to that of RAID 0, it still shows about 1.7 times better I/O performance than performance of single SSD.

The result of this section conveys that the filesystem plays a key role in defining the performance of RAID with SSDs. It also shows insight on decision making for choosing right filesystem for different workloads. The most interesting result shown in the experiments is that F2FS is the choice for the random write with buffered I/O workload, where all other filesystems fail to exhibit better performance than single SSD.

6 Conclusion

In this paper, we used a DELL PERC H710P RAID controller and eight Samsung SSD 840 pro to measure the performance of sequential read/write and random read/write with buffered I/O and direct I/O on various RAID configurations. We find the optimal stripe size to conduct the experiment on given workload, which is found to be 512 KB in our experiments on RAID 0 and RAID 5 with eight SSDs. To analyze the effect of the number of SSDs on the RAID system, we varied the number of the SSDs, and find that the performance of sequential read/write is limited by the performance of RAID controller not by number of SSDs used in the RAID organization. After analyzing the effect of different filesystems on the RAID system, we find that F2FS, the log-structured filesystem, shows the best performance on random write with buffered I/O on RAID 0 and RAID 5 with eight SSDs. The performance of F2FS on random write with buffered I/O on RAID 0 and RAID 5 shows about 5 times and 50 times, respectively.

Acknowledgments This work is supported by IT R&D program MKE/KEIT (No. 10041608, Embedded System Software for New-memory based Smart Device), and supported by a grant from Samsung Electronics Co., Ltd.

References

1. M. Balakrishnan, A. Kadav, V. Prabhakaran, D. Malkhi, Differential raid: rethinking raid for ssd reliability. Trans. Storage **6**(2), 4:1–4:22 (2010)
2. H. Lee, K. Lee, S. Noh, Augmenting raid with an ssd for energy relief. in: *Proceedings of the 2008 Conference on Power Aware Computing and Systems*. pp. 12–12. HotPower'08, USENIX Association (2008)

3. S. Moon, S. Kim, S.W. Lee, Revisiting raid configuration on the flash ssd. Korean Soc. Internet Inf. **9**, 237–242 (2008)
4. D. Narayanan, E. Thereska, A. Donnelly, S. Elnikety, A. Rowstron, Migrating server storage to ssds: analysis of tradeoffs. in: *Proceedings of the 4th ACM European Conference on Computer Systems*. pp. 145–158. ACM (2009)
5. N. Nishikawa, M. Nakano, M. Kitsuregawa, Energy aware raid configuration for large storage systems. in: *Proceedings of the 2011 Green Computing Conference and Workshops*. pp. 1–5 (2011)
6. D. Iacono, M. Shirer, Enterprises look to accelerate applications with all-solid state storage arrays, according to idc's first all flash array forecast (2013), http://www.idc.com/getdoc.jsp?containerId=prUS24072313
7. N. Jeremic, G. Mühl, A. Busse, J. Richling, The pitfalls of deploying solid-state drive raids. in: *Proceedings of the 4th Annual International Conference on Systems and Storage.* pp. 14:1–14:13. SYSTOR '11, ACM (2011)
8. DA. Patterson, G. Gibson, RH. Katz, A case for redundant arrays of inexpensive disks (raid). in: *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*. pp. 109–116. ACM (1988)
9. J. He, J. Bennett, A. Snavely, Dash-io: an empirical study of flash-based io for hpc. in: *Proceedings of the 2010 TeraGrid Conference*. pp. 10:1–10:8. ACM (2010)
10. S. Lee, D. Shin, Performance analysis on ssd raid device. Korea Inf. Sci. Soc. **39**, 367–369 (2012)
11. I. Petrov, G. Almeida, A. Buchmann, U. Gräf, Building large storage based on flash disks. in: ADMS 2010 (2010)
12. A. Mathur, M. Cao, S. Bhattacharya, A. Dilger, A. Tomas, L. Vivier, The new ext4 filesystem: current status and future plans. in: Linux Symposium. vol. 2, pp. 21–33. Citeseer (2007)
13. Wang, R., Anderson, T.: xfs: a wide area mass storage file system. in: 4th Workstation Operating Systems. pp. 71–78 (1993)
14. O. Rodeh, J. Bacik, C. Mason, Btrfs: the linux b-tree filesystem. Trans. Storage **9**(3), 9:1–9:32 (2013)
15. R. Konishi, Y. Amagai, K. Sato, H. Hifumi, S. Kihara, S. Moriai, The linux implementation of a log-structured file system. SIGOPS Oper. Syst. Rev. **40**(3), 102–107 (2006)
16. J. Kim, F2fs (2012), http://www.kernel.org/doc/Documentation/filesystems/f2fs.txt
17. P.M. Chen, E.K. Lee, G.A. Gibson, R.H. Katz, D.A. Patterson, Raid: high-performance, reliable secondary storage. ACM Comput. Surv. **26**(2), 145–185 (1994)
18. S. Jeong, K. Lee, J. Hwang, S. Lee, Y. Won, Framework for analyzing android i/o stack behavior: from generating the workload to analyzing the trace. Future Internet **5**(4), 591–610 (2013)

Evaluation of Fairness in Message Broker System Using Clustered Architecture and Mirrored Queues

Maciej Rostanski, Krzysztof Grochla and Aleksander Seman

Abstract The paper presents a performance evaluation of a message broker system in various high availability configurations. We verify different redundancy architectures against queuing system performance on the example of Rabbit MQ system. We discuss fairness issues and find that the replication of queues may lead to significant differences in the performance offered to clients connected to different nodes of messaging system. Basing on the analysis, we propose cluster architectures which provide fair allocation of resource to multiple clients, while maintaining the redundancy and high performance.

Keywords Mirrored queue · Cluster node · Message broker · High availability · Fault tolerance · Load balancing · Rabbit MQ

1 Introduction

Modern applications, devices, or appliances, being a distributed parts of the whole solution, need to connect and scale. The queuing system that joins multiple servers is necessary component is that distributed the messaging, understood as an information flow or network of application servers [1]. From a designer's perspective, message-oriented middleware can be seen as a:

M. Rostanski (✉)
Academy of Business in Dabrowa Gornicza,
ul. Cieplaka 1C, 41-300 Dabrowa Gornicza, Poland
e-mail: mrostanski@wsb.edu.pl

K. Grochla · A. Seman
Proximetry Poland Sp. z o.o., Al. Rozdzienskiego 91, 40-203 Katowice, Poland
e-mail: kgrochla@proximetry.pl

A. Seman
e-mail: aseman@proximetry.pl

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_42

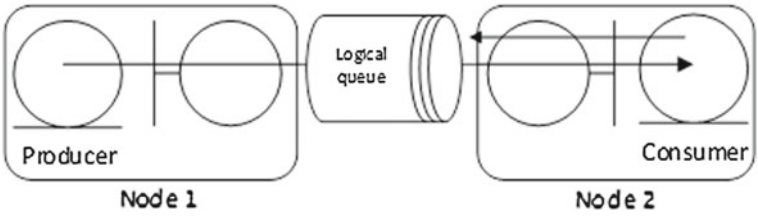


Fig. 1 Message queuing example: messages are stored in FIFO queue. *Source* [3]

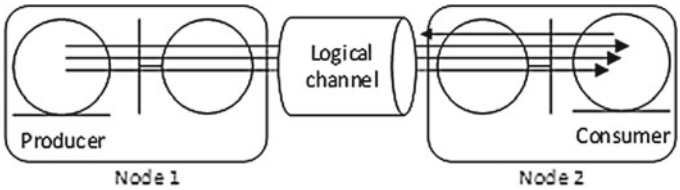


Fig. 2 The publish/subscribe interaction paradigm decouples consumers and producers in terms of space, time, and synchronisation [3]

1. Queuing system, where messages are concurrently pulled by consumers (Fig. 1). Such model is often described as point-to-point queuing. In such case, it is not defined by the element's structure, which element is retrieved by a consumer—it depends on the predesigned order in which the elements are stored in the queue.
2. subscription-based exchange solution, allowing groups of consumers to subscribe to groups of publishers, resulting in a communication network or platform, or a message bus (Fig. 2).

Such bus or queuing system has to be able to scale in terms of geographical distance as well as in terms of devices or applications served. Jones et al. define the role of the queuing system as a distribution point for information sent from the publishers to be distributed to the necessary subscribers and allows applications continue operation while waiting for data from other locations [5].

The middleware layer, often referred to as a 'glue' between different system components, allows communication between them. The modern requirement is to overpower the limits of point-to-point communication, and, moreover, to do it in a nonsynchronous fashion. This is also referred to as a time-, space-, and synchronisation-decoupling [3], and is especially important, given the fact, that the distributed systems now involve thousands of entities, which may be distributed throughout vast geographical distances, and whose behavior and even location may vary in time.

This paper describes the effects of scalability and high availability (HA) improving solutions using RabbitMQ software as a middleware. RabbitMQ is an open source message broker and queuing server that is becoming more and more popular. The design challenge behind this problem are contrary requirements: the scalability and other performance-optimization mechanisms are in principle hindered by high avail-

ability (or even fault tolerance) solutions, which basically put stability and durability over performance.

The paper is organized as follows: the logical components of middleware system are presented and briefly explained specifically, scalability, and high availability solutions are discussed. Next, section describes the experimental topology, evaluation methods, and tools. The main part includes evaluation results for different message broker configuration scenarios for scalability and high availability taking into account the fairness provided to different clients; the experimental results of constructed systems are described and then concluded in the last section.

2 Clustered Queueing Systems Techniques and Solutions Overview

In this paper, we consider a queuing system based on the AMQP protocol and message broker. In the core of the message broker architecture are queues; every message received by the RabbitMQ always is placed in a queue. Messages in queues can be stored in memory (memory-based) or on a disk (disk-based). Second important elements of the RabbitMQ are *exchanges*—the delivery service for messages. The exchange used by a publish operation determines if the delivery will be direct or publish-and-subscribe. A client chooses the exchange used to deliver each message as it is published. The exchange looks at the information in the headers of a message and selects where they should be transferred to. This way AMQP brings the various messaging idioms together—clients can select which exchange should route their messages [6].

The main design concepts revolve around two distinct requirements–that the infrastructure should be (1) scalable and (2) highly available.

Scalability is an architectural characteristic which can be defined as a capability to cope and perform under an increased or expanding workload. A system that scales well will be able to maintain or even increase its level of performance or efficiency when tested by larger operational demands. In terms of message queuing, or even publisher/consumer exchange system, this would mean the possibility of increasing processing speed or message throughput, user capacity, etc.

HA means that the system can deal with its internal failures. Given the availability (A) formula (Eq. 1),

$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} \tag{1}$$

HA aims to minimize downtime and IT service disruption; so the common goal in HA is to increase Mean Time Between Failure (MTBF) and decrease Mean Time to Repair (MTTR). HA applications are designed to have a high level of service uptime. HA solutions may feature many elements, e.g.: system management, live replacement (hot-swap), component redundancy, and failover mechanisms. Common strategy is to avoid single points of failure in the system. This can be difficult, because demands

on such systems include not only ensuring the availability of important data, but also efficient resource sharing of the relatively expensive components.

2.1 Cluster Construction

From the client's perspective, the middleware system is put under one very important requirement–sent messages should not be lost under any circumstance. The system is expected to be operable anytime, which is not HA requirement; it can be understood as a Fault Tolerance. Contrary to HA, which implies a service level in which both planned and unplanned outages do not exceed a small stated value [9], fault-tolerant (FT) systems tend to implement as much component redundancy and mirroring techniques as possible, in order to eliminate system failures completely (this is of course from client's perspective, in fact introducing redundant components will make component failures occur faster) [2]. The reliable system should accept messages, but the delivery is another task–it can be delayed or postponed until failure recovery, so that is not FT requirement after all. But this is valid only if the application can handle the message delay situation.

For the purposes of this paper, both HA and FT solutions were considered:

1. HA (Active/Passive solution)
   in which the downtime of message broker service is expected in case of planned or unplanned unavailability of primary server. Queues and messages have to be persistent (disk-based), and message broker can be restarted elsewhere in the system. Possible solution is to deploy clustering HA solution like pacemaker [7] in order to manage message broker and restart it (or migrate) when necessary using available resources.
2. FT (Active/Active solution)
   means that the planned or unplanned downtime of message broker does not have any effect on queuing system. Typically, it is implemented by MB leveraging clustering mechanism built-in RabbitMQ, which is developed strictly for such situations, and replicates queues on every RabbitMQ node in the cluster [8].

The constructed message-broker cluster is presented on Fig. 3. RabbitMQ nodes are providing services to clients that are load-balanced by external solution (Haproxy). The client emulators publishing or consuming messages were used to evaluate the performance [4]. A four-node cluster was constructed, with nodes capable of putting messages to RAM or disk, depending on test configuration.

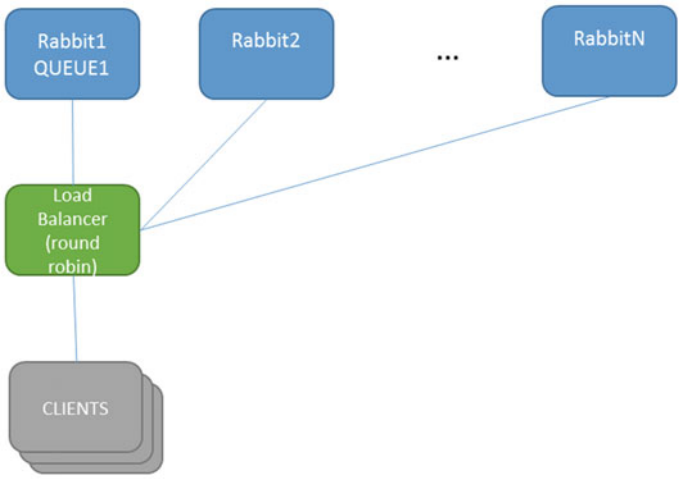


Fig. 3 Cluster structure

2.2 Queues Placement in Cluster

The most interesting design opportunity lies within queues placement. The queue can be *single*, which means it is located only on one node; or it can be *mirrored*, which means more than one node hosts the queue and its messages.

The queue owner node has full information about it; other nodes in the cluster only know the queue's metadata and a pointer to the node where the queue actually is stored. This solution allows to limit storage space requirements and increase performance replicating messages to every node would result in increase of network and disk load for every node, keeping the performance of the cluster the same (or worse) [12]. Regardless where publish is made, message is forwarded to the queue owner node. This leads to main performance-optimization technique: to increase performance for every added node by spreading queues across nodes [11].

On the contrary to performance-driven requirements for queues, there is a need for queue to be redundant when the main goal is to achieve high availability and fault tolerance. If a queue owner node fails, all of the messages within a queue are gone. An active-active redundancy option is possible; any queue can be mirrored. The mirrored queue is achieved by creating slave copies of the queue on other nodes in the cluster. It can be copied on every node, but the designer is able to specify a subset of nodes in the cluster for a queue to live on (Fig. 4).

There are two flow control mechanisms in RabbitMQ. Both work by exerting TCP back pressure on connections that are publishing too fast. They are:

1. A per-connection mechanism that prevents messages being published faster than they can be routed to queues.

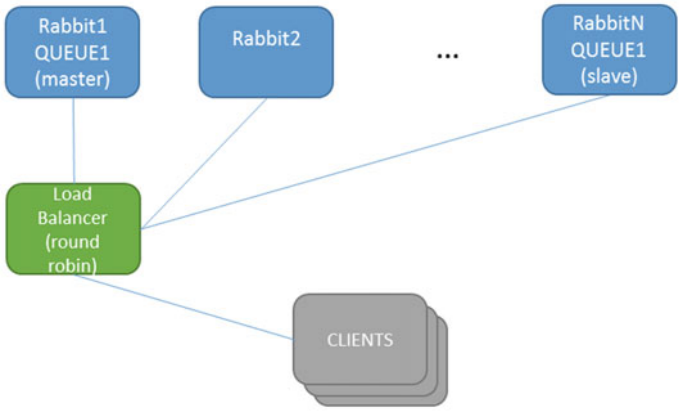


Fig. 4 Queues placement possibilities example

2. A global mechanism that prevents any messages from being published when the memory usage exceeds a configured threshold or free disk space drops below a configured threshold.

Described configuration possibilities create an opportunity to design multiple scenarios, in which specific queues are mirrored to every node, or are mirrored to specific nodes only; the evaluation of these scenarios should give performance results and an answer whether it is possible to create the topology that is scalable, yet maintains High Availability requirements regarding given queue and its messages, as it was described in [10]. Here, we present a detailed analysis of the fairness of resource allocation in the messaging system, and propose two architectures that provide good balance between reliability, fairness, and performance.

3 Extensive Testing Results

3.1 Test Environment

Extensive testing was conducted using different node/queue combinations with or without mirroring. All of scenarios were tested with commodity-equipped virtual machines (single core, 4 GB RAM, 8 GB HDD). The hypervisor host was equipped with Intel i7 CPU and 32 GB RAM. There was no resource overload. Following key factors have been developed in the process:

1. Placement—means whether the queue is hosted
   (a) on the node (“master”),
   (b) is being replicated onto the node (“slave”) or

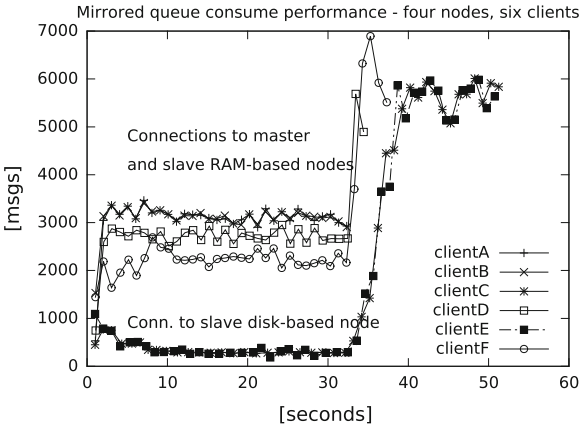


Fig. 5 Performance of message storing in RAM or disk

(c) is not present at this node, but is instead redirected to *master* node, as described at Sect. 2.2—such nodes can be described as "empty" or "other";

2. Whether the given node is storing messages in RAM or disk.

3.2 The Unfairness of Service Operation

The first set of tests have been executed to show the influence of storage configuration on the Rabbit MQ performance. The tests was executed with six clients and four cluster nodes configured. A full-mirror queue was created, which means it resided on one node ("master") and was replicated to every other node ("slave"), but one slave was disk-based instead of RAM. The results can be seen on Fig. 5. Connections to nodes are performing much better when stored in RAM, which is an expected behavior.

The influence of the location of the queues on particular nodes was analyzed in the second test, presented on Fig. 6. This time, every cluster node was storing messages in RAM, but not every node had replicated queue and its messages (means such node was "other" node). Clients connected to nodes without queue turned out to be preferred and thus serviced faster than clients connected directly to "master" or "slave" nodes (where queue is being replicated).

There are two different situations—upper diagram of Fig. 6 shows the queue performance when there are multiple clients sending messages (producers), the second diagram presents the queue performance when the clients are receiving messages (consumers). Sending and receiving message rates vary because of load balancing between cluster nodes—some clients connect with *master* node, some to the *slave* nodes, and some to "empty" or *other* nodes.

The effect of uneven distribution and unfair client treatment is visible in the comparison of results for different number of nodes in the cluster. Increasing the number

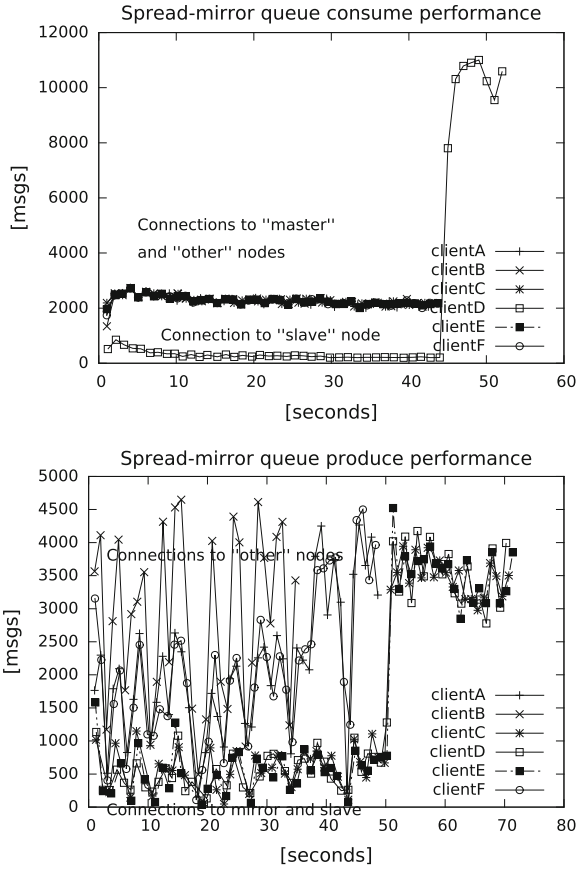


Fig. 6 Examples of the effect of queues placement on the nodes

of nodes results in some performance improvement in overall, but in detailed view it appears that some of the clients have to wait for others to complete, as presented in Fig. 7. The unfairness especially influences consumers (receiving clients), as seen on Fig. 8.

Our tests lead to the conclusions that (1) for sending (producing) rates to be fairly distributed, "master" node should not be serving clients; and (2) if work scenario contains considerably more producers than clients, queues replication to every node is performing better.

3.3 Topologies to Optimize the Fairness

In order to create an environment that should remain fair to every load balanced client, the two topologies shown on Fig. 9 are proposed. The topology on the left

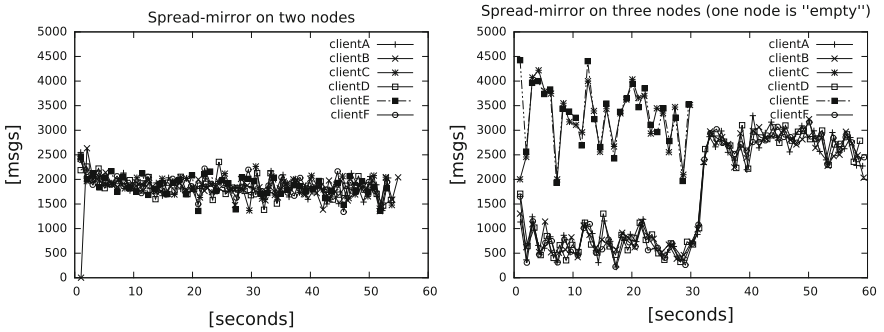


Fig. 7 Message rates of clients sending to "spread" queue in two-node and three-node cluster. The effect of clients hitting "empty" node is visible on the lower diagram

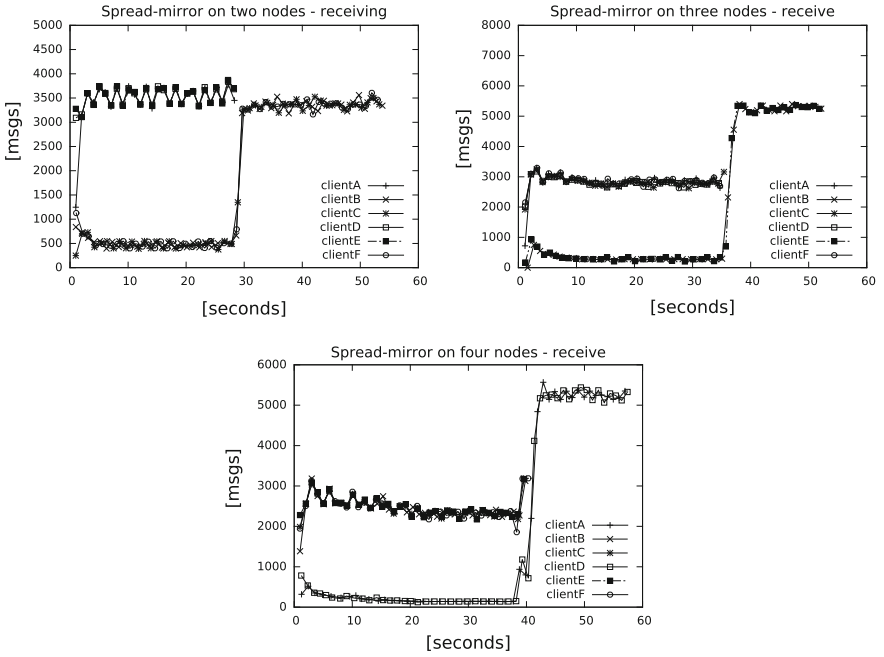


Fig. 8 Message rates for clients receiving from multi-node cluster (two, three and four nodes)

is a fair topology for the queues that are mirrored to every node. This is done by isolating "master" node from clients. For "spread" queues (that are being replicated only to one other node) the fair topology needs one more node, but promises better reliability and scalability. Right side of Fig. 9 shows this concept.

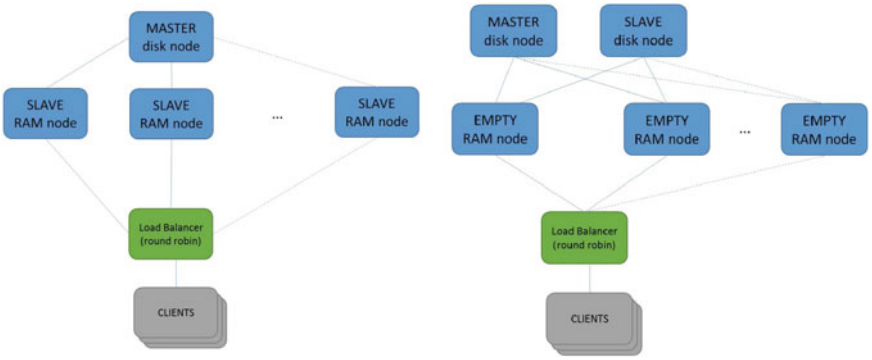


Fig. 9 Proposed topologies to improve fairness for mirrored (*on the left*) and spread (*on the right*) queues system

4 Conclusions

The analysis of the middleware architecture shows that the number of nodes in the cluster and the configuration of data replication amongst them has large influence on the processing speed offered for the clients. The time required for a client to publish or receive message from a analyzed Rabbit MQ system may differ significantly when the client is connected to different nodes in the cluster. The queues have one master node, may be mirrored (to slaves), but not necessarily everywhere (there can be "empty", or "other" nodes). The results show that RabbitMQ-implemented cluster is unfair to clients, depending on which node they get connected to. When sending, "other" nodes are favored, when receiving, "other" nodes and "masters" are favored.

The topology with full-mirrored queues (replicated on every node) is not scalable horizontally (adding more nodes), as our tests show. Based on the available RabbitMQ distribution (3.3.0), the "spread" queues topology did not perform as expected. Message rates were unstable especially for consumers. Erratic behavior of spread queues scenario points to full-mirrored queues as much more stable solution, but "spread" queues scenario gives also promising results.

Acknowledgments This work is partially supported by NCBIR INNOTECH Project K2/HI2/21/1 84126/NCB R/13, The Effective Management of Telecommunication Networks Consist of Millions of Devices.

References

1. M. Altherr, M. Erzberger, S. Maffeis, iBus - a software bus middleware for the Javaplatform, in *Proceedings of the International Workshop on Reliable Middleware Systems*, (1999) pp. 43–53
2. P. Buchwald, The Example of IT System with Fault Tolerance in a Small Business Organization, in *Internet Technical Development and Applications 2*, (Springer, Berlin, 2012), pp. 179–187

3. P.T. Eugster et al., The Many Faces of Publish/Subscribe. ACM Comput. Surv. **35**(2), 114–131 (2003)
4. K. Grochla, L. Naruszewicz, Testing and Scalability Analysis of Network Management Systems Using Device Emulation, in: *Computer Networks* (Springer, Berlinm, 2012), pp. 91–100
5. B. Jones, S. Luxenberg, D. McGrath, P. Trampert and J. Weldon, RabbitMQ Performance and Scalability Analysis, project on CS 4284: Systems and Networking Capstone, Virginia Tech (2011)
6. J. O'Hara, Toward a commodity enterprise middleware. ACM Queue **5**(4), 48–55 (2007)
7. Pacemaker, A scalable High Availability cluster resource manager (2014), http://clusterlabs.org/. Accessed 18 Jan 2014
8. RabbitMQ documentation (2014), http://www.rabbitmq.com/documentation.html. Accessed 21 Jan 2014
9. M. Rostanski, High Availability Methods for Routing in Soho Networks, in *Internet - Technical Developments and Applications 2* (Springer, New York, 2011), pp. 154–152
10. M. Rostanski, K. Grochla, A. Seman, Evaluation of highly available and fault-tolerant middleware clustered architectures using RabbitMQ, in *Proceedings of FEDCSIS* (2014)
11. M. Salvan, A quick message queue benchmark: ActiveMQ, RabbitMQ, HornetQ, QPID, Apollo (2013), http://bit.ly/1b1UGTa
12. A. Videla, J. Williams, *RabbitMQ in Action Distributed messaging for everyone*, Manning (2012)

Erratum to: Information Sciences and Systems 2014

Tadeusz Czachórski, Erol Gelenbe and Ricardo Lent

Erratum to:
T. Czachórski et al., *Information Sciences and Systems 2014*, DOI 10.1007/978-3-319-09465-6

The subtitle of the book was incorrect. The correct book subtitle should read: Proceedings of the 29th International Symposium on Computer and Information Sciences.

The online version of the original book can be found under DOI 10.1007/978-3-319-09465-6

T. Czachórski (✉)
Polish Academy of Sciences, Gliwice, Poland
e-mail: tadek@iitis.gliwice.pl

E. Gelenbe · R. Lent
Imperial College London, London, UK
e-mail: e.gelenbe@imperial.ac.uk

R. Lent
e-mail: r.lent@imperial.ac.uk

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_43

Author Index

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6

Subject Index

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6