

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Ernst Biersack Christian Callegari
Maja Matijasevic (Eds.)

Data Traffic Monitoring and Analysis

From Measurement, Classification,
and Anomaly Detection to Quality of Experience



Springer

Volume Editors

Ernst Biersack
Eurécom, Networking and Security Department
450 Route des Chappes, 06410 Biot, France
E-mail: ernst.biersack@eurecom.fr

Christian Callegari
University of Pisa, Department of Information Engineering
Via Caruso 16, 56122 Pisa, Italy
E-mail: christian.callegari@iet.unipi.it

Maja Matijasevic
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
E-mail: maja.matijasevic@fer.hr

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-36783-0 e-ISBN 978-3-642-36784-7
DOI 10.1007/978-3-642-36784-7
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013933020

CR Subject Classification (1998): C.2.0-6, C.4, D.4.4, A.1, H.3.4-5, H.4.3, H.2.8, K.6.5

LNCS Sublibrary: SL 5 – Computer Communication Networks and Telecommunications

© Springer-Verlag Berlin Heidelberg 2013

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

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

Foreword

Traffic monitoring and analysis (TMA) is an important research topic within the field of communication networks, involving many research groups worldwide that are collectively advancing our understanding of real packet networks and their users. Modern packet networks are highly complex and ever-evolving objects. Understanding, developing, and managing such systems is difficult and expensive in practice. TMA techniques play an increasingly important role in the operation of networks.

Besides its practical importance, TMA is an intellectually attractive research field. First, the inherent complexity of the Internet has attracted many researchers to face traffic measurements since the pioneering times. Second, TMA offers a fertile ground for theoretical and cross-disciplinary research, such as the various analysis techniques being imported into TMA from other fields, while at the same time providing a clear perspective for the exploitation of the results in a real environment. In other words, TMA research has an intrinsic potential to reconcile theoretical investigations with practical applications, and to realign curiosity-driven with problem-driven research.

This book was conceived and prepared in the framework of the COST Action IC0703 titled “Data Traffic Monitoring and Analysis: Theory, Techniques, Tools and Applications for the Future Networks,” or TMA Action for short (see <http://www.tma-portal.eu>).

The COST program is an intergovernmental framework for European Cooperation in Science and Technology, promoting the coordination of nationally funded research on a European level. Each COST Action aims at improving the coordination and exchange between European researchers involved in a particular field or cross-disciplinary topic, and helps to open European Research to cooperation worldwide.

The COST TMA Action was launched in March 2008 and finished in March 2012. It has involved more than 50 research groups from 26 different countries. The primary goal of the TMA Action was the establishment of a recognizable community out of a set of research groups and individuals who work across Europe in the exciting field of Internet traffic monitoring and network measurements. This goal has been largely achieved, and the TMA community that has formed around the Action is now a recognized entity, inside and outside Europe, even after the formal termination of the Action.

The TMA Action launched in 2009 an annual full-day single-track workshop on “Traffic Monitoring and Analysis” with scientific papers selected from an open call through a peer-review process. The TMA Workshop had its fifth edition in 2013, the first one after the closing of the Action, and will continue to provide an occasion for the TMA community to meet at an annual basis.

The TMA Action triggered numerous new research collaborations, typically initiated via research visits (more than 40) and/or joint publications, which will continue in the years to come. Some of these collaborations have been the seed for new international projects in the EU FP7 program. It has organized three full-week PhD schools that involved more than 120 students. The TMA mailing list has over 250 subscribers and keeps growing as new PhD students and young scientists join.

The role of TMA Action as catalyst for “networking researchers” should not obfuscate the “research in networking” achievements. The scientific output of the TMA Action comprises more than 50 joint publications (between two or more research groups in different participating countries) in journals and international conferences, one special issue of a journal, the TMA Workshop proceedings and, of course, this book.

The book is organized into three parts that reflect the most active research topics within the TMA Action. Packet networks will keep evolving in the coming years, new technologies and applications will emerge and user habits will inevitably change. New problems and challenges will rise. But the need to observe and understand, monitor, and analyze the network and its traffic will not vanish. The methodologies and results achieved by the people involved are relevant not only for the present “Internet” as we know it, but also for the “Future Internet” to come. In this sense, this book does not merely bear witness to some piece of past work but represents a useful tool for the future.

Most chapters in this book have been co-authored by experts from different research groups, which emphasizes the collaborative nature of the research activities carried out in the Action.

As initiator and Chair of the TMA Action, I am grateful to all the authors who have contributed to realize this book. Special thanks to the three book editors, Ernst, Christian, and Maja, for their indefatigable work of selection, organization, review, and shepherding of every individual chapter.

I also want to thank the COST Office for supporting the publication of the book.

I wish you a profitable and enjoyable reading!

Fabio Ricciato

Preface

Packet Switching: A Short History

The underlying principle of the Internet is packet switching, as compared to circuit switching used in the “traditional” telephone network. Packet switching was invented about 50 years ago simultaneously by Paul Baran working for the RAND Corp, a think tank of the USA Department of Defense (DoD), and by Donald Davies, working at the National Physical Laboratory in the UK. In packet switching, the data to be transmitted are segmented and each segment is put in a single packet, which also has the full address of the destination in its header. Each packet is routed throughout the network independently of the other packets sent by the same source. The quest for packet switching was, at least in the USA, initially motivated by the needs of the DoD to have a network that is highly resilient to attacks.

In circuit switched telephone network all the intelligence (e.g., for call forwarding, 800 number services) is inside the network and the end device was a very simple telephone handset. In a packet switched network, the situation is exactly the opposite, namely, the network is very simple and the end devices are intelligent: The network consists of routers and links that interconnect the routers. The routers provide a very basic packet-forwarding service: for each packet, the operations at a router are as follows: (a) receive a packet, (b) read the destination address from the packet header, (c) use the destination address to determine over which link to send the packet next. If the outgoing link is currently busy, the packet will temporarily be buffered; if the packet cannot be transmitted and the packet buffer is full, the packet will be simply dropped. As a consequence, a packet switched network only offers a best effort service and leaves it up to the end devices to take the necessary steps to implement the service (e.g., reliable in order data delivery as offered by TCP) required by the application.

When packet switching was invented at the beginning of the 1970s, computers were few and mostly mainframes. It was only some years later that companies such as Digital Equipment Corporation started building the first microcomputers or that Intel was founded. In the early 1980s, workstations were introduced and at about the same time personal computers started to take off when IBM introduced its first PC.

The Evolution of the Internet

When computers started to proliferate, the second premise of packet switching, namely, intelligent end devices that do most of the work was met. Ever since, thanks to Moore's law, the CPU power has been doubling every 18 months. Complex and CPU-intensive operations such as coding and compression of audio or video and forward error correction, all of which were originally implemented in dedicated hardware, could now be realized in software and executed on a standard PC. Only then could the Internet as we know it today develop as:

- The simplicity of packet switching allowed the Internet to span the whole globe as an interconnection of autonomous networks
- Powerful end devices made possible applications such as Skype, video on demand, and all the other innovations that happened at the edge of the network

The separation between (a) the IP network with its best effort transmission and (b) the applications deployed at the edge brought about innovation in terms of new services such as search engines, audio or video transfer, file sharing, or social networks. The use of TCP as a transport protocol turned out to be crucial in assuring a fair sharing of the network resources among all applications. Since network operators no longer control the deployment of new services as they did in the case of the telephone network, their task of network planning and traffic engineering becomes very difficult:

- The advent of Peer-to-Peer, one-click hosting, video-sharing websites, on-demand media streaming, and other services resulted in a dramatic and unforeseen increase of the traffic inside each network and across the peering links interconnecting different networks.
- Content distribution networks and application service providers, which often operate on a global scale to deliver a major part of the today's Internet traffic to the end-users, use their own techniques to balance the load among their sites.
- Viruses infecting millions of computers and spam campaigns contribute unwanted traffic.

Given this state of anarchy, there is a dire need for tools to operate and monitor networks and services in order to detect trends, anomalies, and service degradations.

The Role of Data Traffic Monitoring and Analysis

Traffic monitoring and analysis (TMA) has always been seen as a key methodology to understand telecommunication networks and Internet technology and operation. The evolution and convergence of networks in the last decade have been characterized by dramatic changes to the way users behave, interact, and utilize the network. New categories of applications such as network games, peer-to-peer, and multimedia services have been introduced. Malicious applications

and attacks have become a daily threat to network stability and security. Network service providers aim at both optimizing resource consumption and quality of service, as well as troubleshooting and solving problems, and reducing threats. These topics become even more relevant when considering some characteristics of the nowadays networks, such as the highly distributed nature of network services, the multiplication of threats in terms of spread and sophistication, and the ever increasing bandwidth. The research community is responding to these new challenges by designing innovative algorithms, methodologies, and techniques for TMA. Recent advances in the field of TMA techniques play a key role in the operation of real networks, ensuring their smooth operation as well as the user satisfaction with the quality of services provided over the network. This book is intended to provide the researchers and practitioners with both a review of the state of the art and the most significant contributions in the field of traffic monitoring and analysis.

About the Book

This book was prepared as the Final Publication of COST Action IC0703 “Data Traffic Monitoring and Analysis: Theory, Techniques, Tools and Applications for the Future Networks.” It contains 14 chapters which demonstrate the results, the quality, and the impact of European research in the field of TMA in line with the scientific objective of the Action. We hope it will provide valuable information to researchers as well as practitioners, standards developers, and policy-makers.

The chapters in this book have been collected through an open, but rather selective, two-stage submission and review process. Initially, an open call for contributions was announced among the COST TMA Action participants, and a total of 20 extended abstracts were submitted for consideration in response to the call. Out of these, 17 contributions were selected for full chapter submission. All the submitted contributions were peer-reviewed by three independent reviewers (including some reviewers not directly related to the COST TMA Action), appointed by the editors, and after the first round of reviews, 15 chapters remained. These have been revised according to the reviewers’ comments, some of them substantially (in some cases with the aid of an external shepherd), and resubmitted for the second round of reviews. Finally, 14 chapters were accepted for publication in this book.

The book is structured into three parts. Each part contains general overview chapters, which present the state of the art, followed by chapters focusing on selected problems or presenting traffic measurement techniques and methodologies for a given class of problems. Part I, entitled “Network and Topology Measurement and Modelling,” contains five chapters. Chapters 1, 3, and 4 present the state of the art in the field of high-performance traffic processing and network mapping, while Chaps. 2 and 5 present novel results on available bandwidth estimation techniques and peer-to-peer network analysis, respectively. Part II, entitled “Traffic Classification and Anomaly Detection,” contains four chapters. Chapters 6 and 7 provide surveys in the fields of traffic classification and network anomaly detection, respectively, while the recent advances in anomaly detection

have been further covered in more detail in Chaps. 8 and 9. Part III, entitled “Quality of Experience,” contains five chapters. The introductory chapter, Chap. 10, presents quality of experience (QoE) as a new measurement challenge, while Chap. 11 focus on QoE for YouTube video. Chapters 12 and 13 focus on peer-to-peer IPTV. Chapter 14 deals with mechanisms for QoE driven optimization and enhancement.

The editors wish to thank all the authors, the reviewers, and the shepherds for their excellent work and their responsiveness.

Ernst Biersack
Christian Callegari
Maja Matijasevic

List of Reviewers

Javier Aracil	Universidad Autonoma de Madrid, Spain
Patrik Arlos	Blekinge Institute of Technology, Sweden
Ernst Biersack	EURECOM, France
Pierre Borgnat	ENS, Lyon, France
Alessio Botta	University Federico II of Naples, Italy
Lothar Braun	Technische Universität München, Germany
Christian Callegari	University of Pisa, Italy
Angelo Coluccia	University of Salento, Italy
Alessandro D'Alconzo	Telecommunications Research Center Vienna, Austria
Luca Deri	University of Pisa, Italy
Amogh Dhamdhere	CAIDA University of California San Diego, USA
Jordi Domingo-Pascual	Universitat Politecnica de Catalunya BarcelonaTECH, Spain
Benoit Donnet	Université de Liege, Belgium
Wendy Ellens	TNO, The Netherlands
Francesco Fusco	ETH Zurich, Switzerland
Francesco Gringoli	University of Brescia, Italy
Tobias Hossfeld	University of Würzburg, Germany
Andreas Kassler	Karlstad University, Sweden
Simon Knight	University of Adelaide, Australia
Udo Krieger	University of Bamberg, Germany
Michel Mandjes	University of Amsterdam, The Netherlands
Maja Matijasevic	University of Zagreb, Croatia
Marco Mellia	Politecnico di Torino, Italy
Mu Mu	Lancaster University, UK
Michele Pagano	University of Pisa, Italy
Symeon Papavassiliou	National Technical University of Athens, Greece
Antonio, Pescapé	University Federico II of Naples, Italy
Louis Plissoneau	Orange Labs, France
Gregorio Procissi	University of Pisa, Italy
Nicholas Race	Lancaster University, UK

Fulvio Riso	Politecnico di Torino, Italy
Kamil Sarac	University of Texas at Dallas, USA
Raimund Schatz	Telecommunications Research Center Vienna, Austria
Rene Serral-Gracia	Universitat Politecnica de Catalunya BarcelonaTECH, Spain
Yuval Shavitt	University of Tel Aviv, Israel
Lea Skorin-Kapov	University of Zagreb, Croatia
Dominik Strohmeier	Technische Universität Berlin, Germany
Guillaume Urvoy-Keller	University of Nice, France
Stefan Winkler	Advanced Digital Sciences Center
Sandrine Vaton	Telecom Bretagne, Brest, France

Organization COST

COST - the acronym for European Cooperation in Science and Technology - is the oldest and widest European intergovernmental network for cooperation in research. Established by the Ministerial Conference in November 1971, COST is presently used by the scientific communities of 36 European countries to cooperate in common research projects supported by national funds.

The funds provided by COST - less than 1% of the total value of the projects - support the COST cooperation networks (COST Actions) through which, with EUR 30 million per year, more than 30 000 European scientists are involved in research having a total value which exceeds EUR 2 billion per year. This is the financial worth of the European added value which COST achieves.

A "bottom up approach" (the initiative of launching a COST Action comes from the European scientists themselves), "à la carte participation" (only countries interested in the Action participate), "equality of access" (participation is open also to the scientific communities of countries not belonging to the European Union) and "flexible structure" (easy implementation and light management of the research initiatives) are the main characteristics of COST.

As precursor of advanced multidisciplinary research COST has a very important role for the realisation of the European Research Area (ERA) anticipating and complementing the activities of the Framework Programmes, constituting a "bridge" towards the scientific communities of emerging countries, increasing the mobility of researchers across Europe and fostering the establishment of "Networks of Excellence" in many key scientific domains such as: Biomedicine and Molecular Biosciences; Food and Agriculture; Forests, their Products and Services; Materials, Physical and Nanosciences; Chemistry and Molecular Sciences and Technologies; Earth System Science and Environmental Management; Information and Communication Technologies; Transport and Urban Development; Individuals, Societies, Cultures and Health. It covers basic and more applied research and also addresses issues of pre-normative nature or of societal importance.

Web: <http://www.cost.eu>

Table of Contents

Part I: Network Measurement

High-Performance Network Traffic Processing Systems Using Commodity Hardware	3
<i>José Luis García-Dorado, Felipe Mata, Javier Ramos, Pedro M. Santiago del Río, Victor Moreno, and Javier Aracil</i>	
Active Techniques for Available Bandwidth Estimation: Comparison and Application	28
<i>Alessio Botta, Alan Davy, Brian Meskill, and Giuseppe Aceto</i>	
Internet Topology Discovery	44
<i>Benoit Donnet</i>	
Internet PoP Level Maps	82
<i>Yuval Shavitt and Noa Zilberman</i>	
Analysis of Packet Transmission Processes in Peer-to-Peer Networks by Statistical Inference Methods	104
<i>Natalia M. Markovich and Udo R. Krieger</i>	

Part II: Traffic Classification and Anomaly Detection

Reviewing Traffic Classification	123
<i>Silvio Valenti, Dario Rossi, Alberto Dainotti, Antonio Pescapè, Alessandro Finamore, and Marco Mellia</i>	
A Methodological Overview on Anomaly Detection	148
<i>Christian Callegari, Angelo Coluccia, Alessandro D'Alconzo, Wendy Ellens, Stefano Giordano, Michel Mandjes, Michele Pagano, Teresa Pepe, Fabio Ricciato, and Piotr Żurawski</i>	
Changepoint Detection Techniques for VoIP Traffic	184
<i>Michel Mandjes and Piotr Żurawski</i>	
Distribution-Based Anomaly Detection in Network Traffic	202
<i>Angelo Coluccia, Alessandro D'Alconzo, and Fabio Ricciato</i>	

Part III: Quality of Experience

From Packets to People: Quality of Experience as a New Measurement Challenge	219
<i>Raimund Schatz, Tobias Hoßfeld, Lucjan Janowski, and Sebastian Egger</i>	

Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience	264
<i>Tobias Hoßfeld, Raimund Schatz, Ernst Biersack, and Louis Plissonneau</i>	
Quality Evaluation in Peer-to-Peer IPTV Services	302
<i>Mu Mu, William Knowles, Panagiotis Georgopoulos, Steven Simpson, Eduardo Cerqueira, Nicholas Race, Andreas Mauthe, and David Hutchison</i>	
Cross-Layer FEC-Based Mechanism for Packet Loss Resilient Video Transmission	320
<i>Roger Immich, Eduardo Cerqueira, and Marilia Curado</i>	
Approaches for Utility-Based QoE-Driven Optimization of Network Resource Allocation for Multimedia Services	337
<i>Lea Skorin-Kapov, Krunoslav Ivesic, Giorgos Aristomenopoulos, and Symeon Papavassiliou</i>	
Author Index	359

Part I

Network Measurement

High-Performance Network Traffic Processing Systems Using Commodity Hardware

José Luis García-Dorado, Felipe Mata, Javier Ramos,
Pedro M. Santiago del Río, Victor Moreno, and Javier Aracil

High Performance Computing and Networking,
Universidad Autónoma de Madrid,
Madrid, Spain

Abstract. The Internet has opened new avenues for information accessing and sharing in a variety of media formats. Such popularity has resulted in an increase of the amount of resources consumed in backbone links, whose capacities have witnessed numerous upgrades to cope with the ever-increasing demand for bandwidth. Consequently, network traffic processing at today's data transmission rates is a very demanding task, which has been traditionally accomplished by means of specialized hardware tailored to specific tasks. However, such approaches lack either of flexibility or extensibility—or both. As an alternative, the research community has pointed to the utilization of commodity hardware, which may provide flexible and extensible cost-aware solutions, ergo entailing large reductions of the operational and capital expenditure investments. In this chapter, we provide a survey-like introduction to high-performance network traffic processing using commodity hardware. We present the required background to understand the different solutions proposed in the literature to achieve high-speed lossless packet capture, which are reviewed and compared.

Keywords: commodity hardware, packet capture engine, high-performance networking, network traffic monitoring.

1 Introduction

Leveraging on the widespread availability of broadband access, the Internet has opened new avenues for information accessing and sharing in a variety of media formats. Such popularity has resulted in an increase of the amount of resources consumed in backbone links, whose capacities have witnessed numerous upgrades to cope with the ever-increasing demand for bandwidth. In addition, the Internet customers have obtained a strong position in the market, which has forced network operators to invest large amounts of money in traffic monitoring on attempts to guarantee the satisfaction of their customers—which may eventually entail a growth in operators' market share.

Nevertheless, keeping pace with such ever-increasing data transmission rates is a very demanding task, even if the applications built on top of a monitoring

system solely capture to disk the headers of the traversing packets, without further processing them. For instance, traffic monitoring at rates ranging from 100 Mb/s to 1 Gb/s was considered very challenging a few years ago, whereas contemporary commercial routers typically feature 10 Gb/s interfaces, reaching aggregated rates as high as 100 Tb/s.

As a consequence, network operators have entrusted to specialized Hardware (HW) devices (such as FPGA-based solutions, network processors or high-end commercial solutions) with their networks monitoring. These solutions are tailored to specific tasks of network monitoring, thus achieving a high-grade of performance—e.g., lossless capture. However, these alternatives for network traffic monitoring either lack of flexibility or extensibility (or both), which are mandatory nowadays for large-scale network monitoring. As a result, there have been some initiatives to provide some extra functionalities in network elements through supported Application Programming Interfaces (APIs) to allow the extension of the software part of their products—e.g., OpenFlow [1].

As an alternative, the research community has pointed to the utilization of commodity hardware based solutions [2]. Leveraging on commodity hardware to build network monitoring applications brings along several advantages when compared to commercial solutions, among which overhang the flexibility to adapt any network operation and management tasks (as well as to make the network maintenance easier), and the economies of scale of large-volume manufacturing in the PC-based ecosystem, ergo entailing large reductions of the operational and capital expenditures (OPEX and CAPEX) investments, respectively. To illustrate this, we highlight the special interest that software routers have recently awakened [3, 4]. Furthermore, the utilization of commodity hardware presents other advantages such as using energy-saving policies already implemented in PCs, and better availability of hardware/software updates enhancing extensibility [4].

To develop a network monitoring solution using commodity hardware, the first step is to optimize the default NIC driver to guarantee that the high-speed incoming packet stream is captured lossless. The main problem, the receive live-lock, was studied and solved several years ago [5, 6]. The problem appears when, due to heavy network load, the system collapses because all its resources are destined to serve the per packet interrupts. The solution, which mitigates the interrupts in case of heavy network load, is now implemented in modern operating systems (Section 2.2), and specifically in the GNU Linux distribution, which we take in this chapter as the leading example to present the several capture engines proposed in the literature [4, 7–10]. These engines provide improvements at different levels of the networking stack, mainly at the driver and kernel levels. Another important point is the integration of the capturing capabilities with memory and CPU affinity aware techniques, which increase performance by enhancing the process locality.

The rest of the chapter is structured as follows. In Section 2 we provide the required background to understand the possibilities and limitations that contemporary commodity hardware provides for high-performance networking tasks.

Then, Section 3 describes the general solutions proposed to overcome such limitations. Section 4 details different packet capture engines proposed in the literature, as well as their performance evaluation and discussion. Finally, Section 5 concludes the chapter.

2 Background

2.1 Commodity Hardware: Low Cost, Flexibility and Scalability

Commodity computers are systems characterized by sharing a base instruction set and architecture (memory, I/O map and expansion capability) common to many different models, containing industry-standard PCI slots for expansion that enables a high degree of mechanical compatibility, and whose software is widely available off-the-self. These characteristics play a special role in the economies of scale of the commodity computer ecosystem, allowing large-volume manufacturing with low costs per unit. Furthermore, with the recent development of multi-core CPUs and off-the-self NICs, these computers may be used to capture and process network traffic at near wire-speed with little or no packet losses in 10 GbE networks [4].

On the one hand, the number of CPU cores within a single processor is continuously increasing, and nowadays it is common to find quad-core processors in commodity computers—and even several eight-core processors in commodity servers. On the other hand, modern NICs have also evolved significantly in the recent years calling both former capturing paradigms and hardware designs into question. One example of this evolution is Receive Side Scaling (RSS) technology developed by Intel [11] and Microsoft [12]. RSS allows NICs to distribute the network traffic load among different cores of a multi-core system, overcoming the bottleneck produced by single-core based processing and optimizing cache utilization. Specifically, RSS distributes traffic to different receive queues by means of a hash value, calculated over certain configurable fields of received packets and an indirection table. Each receive queue may be bound to different cores, thus balancing load across system resources.

As shown in Fig. 1, the Least Significant Bits (LSB) from the calculated hash are used as a key to access to an indirection table position. Such indirection table contains values used to assign the received data to a specific processing core. The standard hash function is a Toeplitz hash whose pseudocode is showed in Algorithm 1. The inputs for the function are: an array with the data to hash and a secret 40-byte key (K)—essentially a bitmask. The data array involves the following fields: IPv4/IPv6 source and destination addresses; TCP/UDP source and destination ports; and, optionally, IPv6 extension headers. The default secret key produces a hash that distributes traffic to each queue maintaining unidirectional flow-level coherency—packets containing same source and destination addresses and source and destination ports will be delivered to the same processing core. This behavior can be changed by modifying the secret key to distribute traffic based on other features. For example in [13] a solution for maintaining bidirectional flow-level (session-level) coherency is shown.

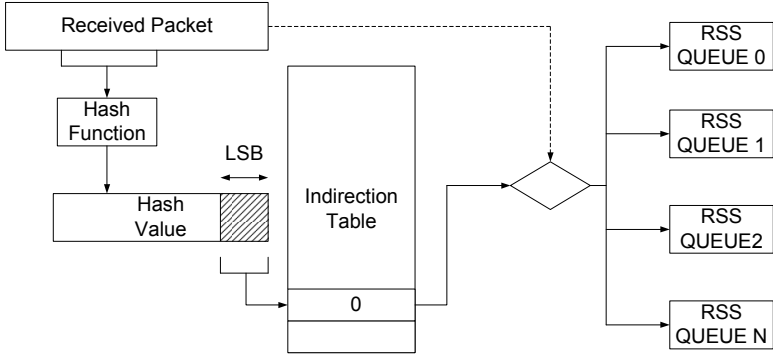


Fig. 1. RSS architecture

Modern NICs offer further features in addition to RSS technology. For example, advanced hardware filters can be programmed in Intel 10 GbE cards to distribute traffic to different cores based on rules. This functionality is called Flow Director and allows the NIC to filter packets by: Source/destination addresses and ports; Type Of Service value from IP header; Level 3 and 4 protocols; and, Vlan value and Ethertype.

Hardware/software interactions are also of paramount importance in commodity hardware systems. For example Non-Uniform Memory Access (NUMA) design has become the reference for multiprocessor architectures, and has been extensively used in high-speed traffic capturing and processing. In more detail, NUMA design splits available system memory between different Symmetric MultiProcessors (SMPs) assigning a memory chunk to each of them. The combination of a processor and a memory chunk is called NUMA node. Fig. 2 shows some examples of NUMA architectures. NUMA-memory distribution boosts up systems' performance as each processor can access in parallel to its own chunk of memory, reducing the CPU data starvation problem. Although NUMA architectures increase the performance in terms of cache misses and memory accesses [14], processes must be carefully scheduled to use the memory owned by the core in which they are being executed, avoiding accessing to other NUMA nodes.

Essentially, the accesses from a core to its corresponding memory chunk results in a low data fetching latency, whereas accessing to other memory chunk

Algorithm 1. Toeplitz standard algorithm

```

1: function COMPUTEHASH(input[],K)
2:   result = 0
3:   for each bit b in input[] from left to right do
4:     if b == 1 then
5:       result ^= left-most 32 bits of K
6:       shift K left 1 bit position
7:   return result

```

increases this latency. To explode NUMA architectures, the NUMA-node distribution must be previously known as it varies across different hardware platforms. Using the `numactl`¹ utility a NUMA-node distance matrix may be obtained. This matrix represents the distance from each NUMA-node memory bank to the others. Thus, the higher the distance is, the higher the access latency to other NUMA nodes is. Other key aspect to get the most of NUMA systems is the interconnection between the different devices and the processors.

Generally, in a traffic capture scenario, NICs are connected to processors by means of PCI-Express (PCIe) buses. Depending on the used motherboard in the commodity hardware capture system, several interconnection patterns are possible. Fig. 2 shows the most likely to find schemes on actual motherboards. Specifically Fig. 2a shows an asymmetric architecture with all PCIe lines directly connected to a processor whereas Fig. 2b shows a symmetric scheme where PCIe lines are distributed among two processors. Figs. 2c and 2d show similar architectures with the difference of having their PCIe lines connected to one or several IO-hubs. IO-hubs not only connect PCIe buses but also USB or PCI buses as well as other devices with the consequent problem of sharing the bus between the IO-hub and the processor among different devices. All this aspects must be taken into account when setting up a capture system. For example, when a NIC is connected to PCIe assigned to a NUMA node, capturing threads must be executed on the corresponding cores of that NUMA node. Assigning capture threads to another NUMA node implies data transmission between processors using Processor Interconnection Bus which leads to performance degradation. One important implication of this fact is that having more capture threads than existing cores in a NUMA node may be not a good approach as data transmission between processors will exist. To obtain information about the assignment of a PCIe device to a processor, the following command can be executed on Linux systems `cat /sys/bus/pci/devices/PCI_ID/local_cpulist` where `PCI_ID` is the device identifier obtained by executing `lspci`² command.

All the previously mentioned characteristics make modern commodity computers highly attractive for high-speed network traffic monitoring, because their performance may be compared to today's specialized hardware, such as FPGAs (NetFPGA³, Endace DAG cards⁴), network processors^{5,6,7} or commercial solutions provided by router vendors⁸, but they can be obtained at significantly lower prices, thus providing cost-aware solutions. Moreover, as the monitoring functionality is developed at user-level, commodity hardware-based solutions are largely flexible, which in addition to the mechanical compatibility, allows

¹ linux.die.net/man/8/numactl

² linux.die.net/man/8/lspci

³ www.netfpga.org

⁴ www.endace.com/

⁵ www.alcatel-lucent.com/fp3/

⁶ www.lsi.com/products/networkingcomponents/Pages/networkprocessors.aspx

⁷ www.intel.com/p/en_US/embedded/hwsw/hardware/ixp-4xx

⁸ www.cisco.com/go/nam

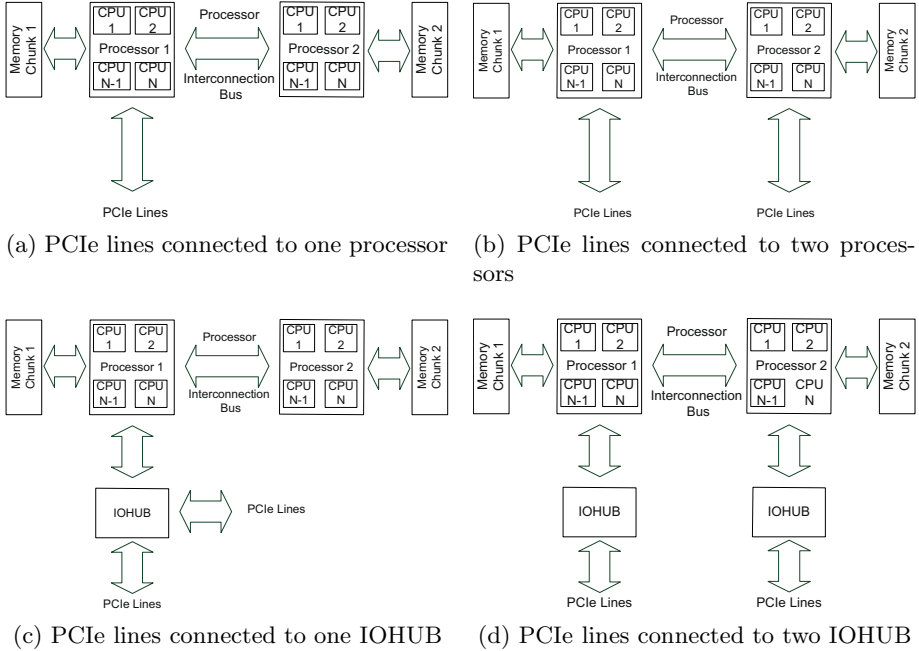


Fig. 2. NUMA architectures

designing scalable and extensible systems that are of paramount importance for the monitoring of large-scale networks.

2.2 Operating System Network Stack

Nowadays, network hardware is rapidly evolving for high-speed packet capturing but software is not following this trend. In fact, most commonly used operating systems provide a general network stack that prioritizes compatibility rather than performance. Modern operating systems feature a complete network stack that is in charge of providing a simple socket user-level interface for sending/receiving data and handling a wide variety of protocols and hardware. However, this interface does not perform optimally when trying to capture traffic at high speed.

Specifically, Linux network stack in kernels previous to 2.6 followed an interrupt-driven basis. Let us explain its behavior: each time a new packet arrives into the corresponding NIC, this packet is attached to a descriptor in a NIC's receiving (RX) queue. Such queues are typically circular and are referred as rings. This packet descriptor contains information regarding the memory region address where the incoming packet will be copied via a Direct Memory Access (DMA) transfer. When it comes to packet transmission, the DMA transfers are made in the opposite direction and the interrupt line is raised once such transfer has been completed so new packets can be transmitted. This mechanism is shared

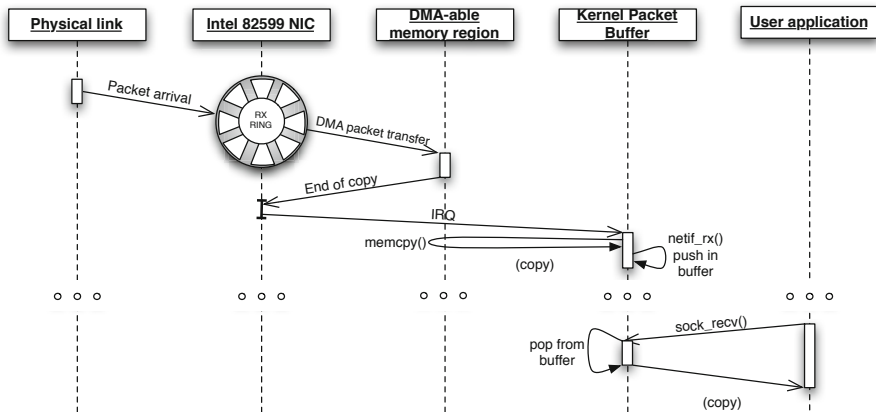


Fig. 3. Linux Network Stack RX scheme in kernels previous to 2.6

by all the different packet I/O existing solutions using commodity hardware. The way in which the traditional Linux network stack works is shown in Fig. 3. Each time a packet RX interrupt is raised, the corresponding interrupt software routine is launched and copies the packet from the memory area in which the DMA transfer left the packet, DMA-able memory region, into a local kernel `sk_buff` structure—typically, referred as packet kernel buffer. Once that copy is made, the corresponding packet descriptor is released (then the NIC can use it to receive new packets) and the `sk_buff` structure with the just received packet data is pushed into the system network stack so that user applications can feed from it. The key point in such packet I/O scheme is the need to raise an interrupt every time a packet is received or transferred, thus overloading the host system when the network load is high [15].

With the aim of overcoming such problem, most current high-speed network drivers make use of the NAPI (New API)⁹ approach. This feature was incorporated in kernel 2.6 to improve packet processing on high-speed environments. NAPI contributes to packet capture speedup following two principles:

- (i) *Interrupt mitigation.* Receiving traffic at high speed using the traditional scheme generates numerous interrupts per second. Handling these interrupts might lead to a processor overload and therefore performance degradation. To deal with this problem, when a packet RX/TX interrupt arrives, the NAPI-aware driver interrupt routine is launched but, differently from the traditional approach, instead of directly copying and queuing the packet the interrupt routine schedules the execution of a `poll()` function, and disables future similar interrupts. Such function will check if there are any packets available, and copies and enqueues them into the network stack if ready, without waiting to an interruption. After that, the

⁹ www.linuxfoundation.org/collaborate/workgroups/networking/napi

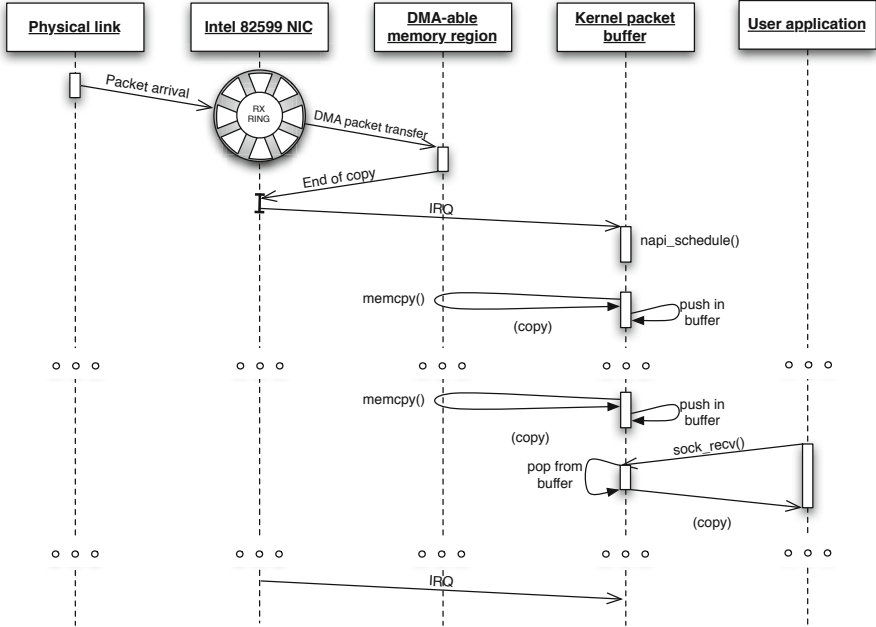


Fig. 4. Linux NAPI RX scheme

same `poll()` function will reschedule itself to be executed in a short future (that is, without waiting to an interruption) until no more packets are available. If such condition is met, the corresponding packet interrupt is activated again. Polling mode is more CPU consumer than interrupt-driven when the network load is low, but its efficiency increases as speed grows. NAPI compliant drivers adapt themselves to the network load to increase performance on each situation dynamically. Such behavior is represented in Fig. 4.

- (ii) *Packet throttling.* Whenever high-speed traffic overwhelms the system capacity, packets must be dropped. Previous non-NAPI drivers dropped these packets in kernel-level, wasting efforts in communication and copies between drivers and kernel. NAPI compliant drivers can drop traffic in the network adapter by means of flow-control mechanisms, avoiding unnecessary work.

From now on, the GNU Linux NAPI mechanism will be used as the leading example to illustrate the performance problems and limitations as it is a widely used open-source operating system which makes performance analysis easier and code instrumentation possible for timing statistics gathering. Furthermore, the majority of the existing proposals in the literature are tailored to different flavors of the GNU Linux distribution. Some of these proposals have additionally paid

attention to other operating systems, for example, FreeBSD [8], but none of them have ignored GNU Linux.

2.3 Packet Capturing Limitations: Wasting the Potential Performance

NAPI technique by itself is not enough to overcome the challenging task of very high-speed traffic capturing since other architectural inherent problems degrades the performance. After extensive code analysis and performance tests, several main problems have been identified [4, 8, 16, 17]:

- (i) *Per-packet allocation and deallocation of resources.* Every time a packet arrives to a NIC, a packet descriptor is allocated to store packet's information and header. Whenever the packet has been delivered to user-level, its descriptor is released. This process of allocation and deallocation generates a significant overhead in terms of time especially when receiving at high packet rates—as high as 14.88 Million packets per second (Mpps) in 10 GbE. Additionally, the `sk_buff` data structure is large because it comprises information from many protocols in several layers, when the most of such information is not necessary for numerous networking tasks. As shown in [16], `sk_buff` conversion and allocation consume near 1200 CPU cycles per packet, while buffer release needs 1100 cycles. Indeed, `sk_buff`-related operations consume 63% of the CPU usage in the reception process of a single 64B sized packet [4].
- (ii) *Serialized access to traffic.* Modern NICs include multiple HW RSS queues that can distribute the traffic using a hardware-based hash function applied to the packet 5-tuple (Section 2.1). Using this technology, the capture process may be parallelized since each RSS queue can be mapped to a specific core, and as a result the corresponding NAPI thread, which is core-bound, gathers the packets. At this point all the capture process has been parallelized. The problem comes at the upper layers, as the GNU Linux network stack merges all packets at a single point at network and transport layers for their analysis. Fig. 5 shows the architecture of the standard GNU Linux network stack. Therefore, there are two problems caused by this fact that degrade the system's performance: first, all traffic is merged in a single point, creating a bottleneck; second, a user process is not able to receive traffic from a single RSS queue. Thus, we cannot make the most of parallel capabilities of modern NICs delivered to a specific queue associated with a socket descriptor. This process of serialization when distributing traffic at user-level degrades the system's performance, since the obtained speedup at driver-level is lost. Additionally, merging traffic from different queues may entail packet disordering [18].
- (iii) *Multiple data copies from driver to user-level.* Since packets are transferred by a DMA transaction until they are received from an application in user-level, packets are copied several times, at least twice: from the DMA-able memory region in the driver to a packet buffer in kernel-level, and from

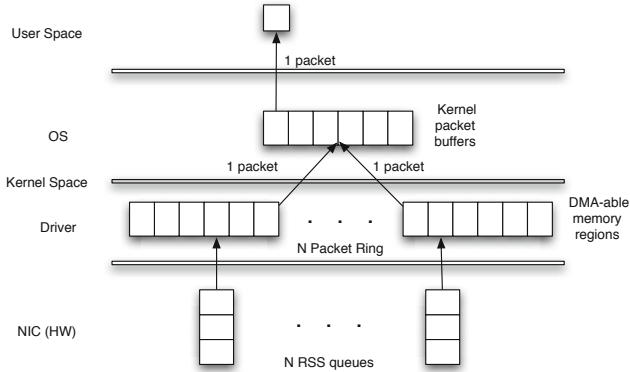


Fig. 5. Standard Linux Network Stack

the kernel packet buffer to the application in user-level. For instance, a single data copy consumes between 500 and 2000 cycles depending on the packet length [16]. Another important idea related to data copy is the fact that copying data packet-by-packet is not efficient, so much the worse when packets are small. This is caused by the constant overhead inserted on each copy operation, giving advantage to large data copies.

- (iv) *Kernel-to-userspace context switching.* From the monitoring application in user-level is needed to perform a system call for each packet reception. Each system call entails a context switch, from user-level to kernel-level and vice versa, and the consequent CPU time consumption. Such system calls and context switches may consume up to 1000 CPU cycles per-packet [16].
- (v) *No exploitation of memory locality.* The first access to a DMA-able memory region entails cache misses because DMA transactions invalidate cache lines. Such cache misses represent 13.8% out of the total CPU cycles consumed in the reception of a single 64B packet [4]. Additionally, as previously explained, in a NUMA-based system the latency of a memory access depends on the memory node accessed. Thus, an inefficient memory location may entail a performance degradation due to cache misses and greater memory access latencies.

3 Proposed Techniques to Overcome Limitations

In the previous sections, we have shown that modern NICs are a great alternative to specialized hardware for network traffic processing tasks at high speed. However, both the networking stack of current operating systems and applications at user-level do not properly exploit these new features. In this section, we present several proposed techniques to overcome the previous described limitations in the default operating systems' networking stack.

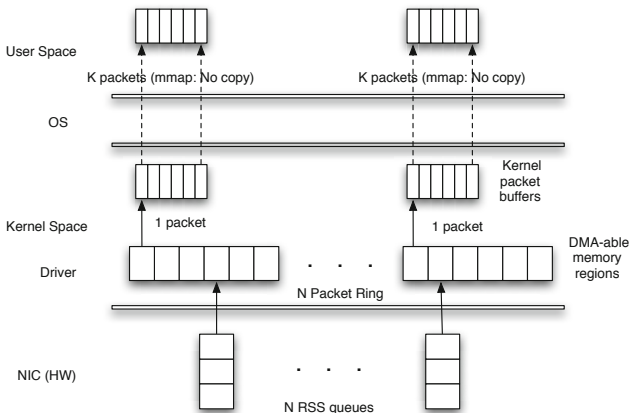


Fig. 6. Optimized Linux Network Stack

Such techniques may be applied either at driver-level, kernel-level or between kernel-level and user-level, specifically applied at the data they exchange, as will be explained.

- (i) *Pre-allocation and re-use of memory resources.* This technique consists in allocating all memory resources required to store incoming packets, i.e., data and metadata (packet descriptors), before starting packet reception. Particularly, N rings of descriptors (one per HW queue and device) are allocated when the network driver is loaded. Note that some extra time is needed at driver loading time but per-packet allocation overhead is substantially reduced. Likewise, when a packet has been transferred to user-space, its corresponding packet descriptor is not released to the system, but it is re-used to store new incoming packets. Thanks to this strategy, the bottleneck produced by per-packet allocation/deallocation is removed. Additionally, `sk_buff` data structures may be simplified reducing memory requirements. These techniques must be applied at driver-level.
- (ii) *Parallel direct paths.* To solve serialization in the access to traffic, direct parallel paths between RSS queues and applications are required. This method, shown in Fig. 6, achieves the best performance when a specific core is assigned both for taking packets from RSS queues and forwarding them to the user-level. This architecture also increases the scalability, because new parallel paths may be created on driver module insertion as the number of cores and RSS queues grow. In order to obtain parallel direct paths, we have to modify the data exchange mechanism between kernel-level and user-level.

In the downside, such technique entails mainly three drawbacks. First, it requires the use of several cores for capturing purposes, cores that otherwise may be used for other tasks. Second, packets may arrive potentially

out-of-order at user-level which may affect some kind of applications [18]. Third, RSS distributes traffic to each receive queue by means of a hash function. When there is no interaction between packets, they can be analyzed independently, which allows to take the most of the parallelism by creating and linking one or several instances of a process to each capture core. However, some networking tasks require analyzing related packet, flows or sessions. For example, a Voice over IP (VoIP) monitoring system, assuming that such a system is based on the SIP protocol, requires not only to monitor the signaling traffic (i.e., SIP packets) but also calls themselves—typically, RTP traffic. Obviously, SIP and RTP flows may not share either level 3 or 4 header fields that the hash function uses to distribute packets to each queue, hence they might be assigned to different queues and cores. The approach to circumvent this latter problem is that the capture system performs by itself some aggregation task. The idea is that before packets are forwarded to user-space (for example to a socket queue), a block of the capture system aggregates the traffic according to a given metric. However, this for sure is at the expense of performance.

- (iii) *Memory mapping.* Using this method, a standard application can map kernel memory regions, reading and writing them without intermediate copies. In this manner, we may map the DMA-able memory region where the NIC directly accesses. In such case, this technique is called zero-copy. As an inconvenient, exposing NIC rings and registers may entail risks for the stability of the system [8]. However, this is considered a minor issue as typically the provided APIs protect NIC from incorrect access. In fact, all video boards use equivalent memory mapping techniques without major concerns. Another alternative is mapping the kernel packet memory region where driver copies packets from RX rings, to user-level, thus user applications access to packets without this additional copy. Such alternative removes one out of two copies in the default network stack. This technique is implemented on current GNU Linux as a standard raw socket with `RX_RING/TX_RING` socket option. Applying this method requires either driver-level or kernel-level modifications and in the data exchange mechanism between kernel-level and user-level.
- (iv) *Batch processing.* To gain performance and avoid the degradation related with per-packet copies, batch packet processing may be applied. This solution groups packets into a buffer and copies them to kernel/user memory in groups called batches. Applying this technique permits to reduce the number of system calls and the consequent context switchings, and mitigates the number of copies. Thus, the overhead of processing and copying packets individually is removed. According to NAPI architecture, there are intuitively two points to use batches, first if packets are being asked in a polling policy, the engines may ask for more than one packet per request. Alternatively, if the packet fetcher works on a interrupt-driven basis, one intermediate buffer may serve to collect traffic until applications ask for it. The major problem of batching techniques is the increase of latency and jitter, and timestamp inaccuracy on received packets because packets have

to wait until a batch is full or a timer expires [19]. In order to implement batch processing, we must modify the data exchange between kernel-level and user-level.

- (v) *Affinity and prefetching.* To increase performance and exploit memory locality, a process must allocate memory in a chunk assigned to the processor in which it is executing. This technique is called memory affinity. Other software considerations are CPU and interrupt affinities. CPU affinity is a technique that allows fixing the execution localization in terms of processors and cores of a given process (process affinity) or thread (thread affinity). The former action may be performed using Linux `taskset`¹⁰ utility, and the latter by means of `pthread_setaffinity_np`¹¹ function of the POSIX pthread library. At kernel and driver levels, software and hardware interrupts can be handled by specific cores or processors using this same approach. This is known as interrupt affinity and may be accomplished writing a binary mask to `/proc/irq/IRQ#/smp_affinity`. The importance of setting capture threads and interrupts to the same core lies in the exploitation of cache data and load distribution across cores. Whenever a thread wants to access to the received data, it is more likely to find them in a local cache if previously these data have been received by an interrupt handler assigned to the same core. This feature in combination with the previously commented memory locality optimizes data reception, making the most of the available resources of a system.

Another affinity issue that must be taken into account is to map the capture threads to the NUMA node attached to the PCIe slot where the NIC has been plugged. To accomplish such task, the system information provided by the `sysctl` interface (shown in Section 2.1) may result useful. Additionally, in order to eliminate the inherent cache misses, the driver may prefetch the next packet (both packet data and packet descriptor) while the current packet is being processed. The idea behind prefetching is to load the memory locations that will be potentially used in a near future in processor's cache in order to access them faster when required. Some drivers, such as Intel `ixgbe`, apply several prefetching strategies to improve performance. Thus, any capture engine making use of such vanilla driver, will see its performance benefited from the use of prefetching. Further studies such as [4, 20] have shown that more aggressive prefetching and caching strategies may boost network throughput performance.

4 Capture Engines Implementations

In what follows, we present four capture engine proposals, namely: PF_RING DNA [10], PacketShader [4], Netmap [8] and PFQ [9], which have achieved significant performance. For each engine, we describe the system architecture (remarking differences with the other proposals), the above-mentioned techniques

¹⁰ linux.die.net/man/1/taskset

¹¹ linux.die.net/man/3/pthread_setaffinity_np

that applies, what API is provided for clients to develop applications, and what additional functionality it offers. Table 1 shows a summary of the comparison of the proposals under study. We do not include some capture engines, previously proposed in the literature, because they are obsolete or unable to be installed in current kernel versions (Routebricks [3], UIO-IXGBE [21]) or there is a new version of such proposals (PF_RING TNAPI [7]). Finally, we discuss the performance evaluation results, highlight the advantages and drawbacks of each capture engine and give guidelines to the research community in order to choose the more suitable capture system.

Table 1. Comparison of the four proposals (D=Driver, K=Kernel, K-U=Kernel-User interface)

Characteristics/ Techniques	PF_RING DNA	PacketShader	netmap	PFQ
Memory Pre-allocation and re-use	✓	✓	✓	×/✓
Parallel direct paths	✓	✓	✓	✓
Memory mapping	✓	✓	✓	✓
Zero-copy	✓	×	×	×
Batch processing	×	✓	✓	✓
CPU and interrupt affinity	✓	✓	✓	✓
Memory affinity	✓	✓	×	✓
Aggressive Prefetching	×	✓	×	×
Level modifications	D,K, K-U	D, K-U	D,K, K-U	D (minimal), K,K-U
API	Libpcap-like	Custom	Standard libc	Socket-like

4.1 PF_RING DNA

PF_RING Direct NIC Access (DNA) is a framework and engine to capture packets based on Intel 1/10 Gb/s cards. This engine implements pre-allocation and re-use of memory in all its processes, both RX and PF_RING queue allocations. PF_RING DNA also allows building parallel paths from hardware receive queues to user processes, that is, it allows to assign a CPU core to each received queue whose memory can be allocated observing NUMA nodes, permitting the exploitation of memory affinity techniques.

Differently from the other proposals, it implements full zero-copy, that is, PF_RING DNA maps user-space memory into the DMA-able memory region of the driver allowing users' applications to directly access to card registers and data in a DNA fashion. In such a way, it avoids the intermediation of the kernel packet buffer reducing the number of copies. However, as previously noted, this is at the expense of a slight weakness to errors from users' applications that occasionally do not follow the PF_RING DNA API (which explicitly does not allow incorrect memory accesses), which may potentially entail system crashes. In the rest of the proposals, direct accesses to the NIC are protected. PF_RING DNA behavior is shown in Fig. 7, where some NAPI steps have been replaced by a zero-copy technique.

PF_RING DNA API provides a set of functions for opening devices to capture packets. It works as follows: first, the application must be registered with `pfring_set_application_name()` and before receiving packets, the reception socket can be configured with several functions, such as, `pfring_set_direction()`, `pfring_set_socket_mode()` or `pfring_set_poll_duration()`. Once the socket is configured, it is enabled for reception with `pfring_enable_ring()`. After the initialization process, each time a user wants to receive data `pfring_recv()` function is called. Finally, when the user finishes capturing traffic `pfring_shutdown()` and `pfring_close()` functions are called. This process is replicated for each receive queue.

As one of the major advantages of this solution, PF_RING API comes with a wrapping to the above-mentioned functions that provides large flexibility and ease of use, essentially following the de facto standard of the libpcap library. Additionally, the API provides functions for applying filtering rules (for example, BPF filters), network bridging, and IP reassembly. PF_RING DNA and a user library for packet processing are free-available for the research community¹².

4.2 PacketShader

The authors of PacketShader (PS) developed their own capture engine to highly optimize the traffic capture module as a first step in the process of developing a software router able to work at multi-10Gb/s rates. However, all their efforts are applicable to any generic task that involves capturing and processing packets. They apply memory pre-allocation and re-use, specifically, two memory regions are allocated—one for the packet data, and another for its metadata. Each buffer has fixed-size cells corresponding to one packet. The size for each cell of packet data is aligned to 2048 bytes, which corresponds to the next highest power of two for the standard Ethernet MTU. Metadata structures are compacted from 208 bytes to only 8 bytes (96%) removing unnecessary fields for many networking tasks.

Additionally, PS implements memory mapping, thus allowing users to access to the local kernel packet buffers avoiding unnecessary copies. In this regard,

¹² www.ntop.org/products/pf_ring/libzero-for-dna/

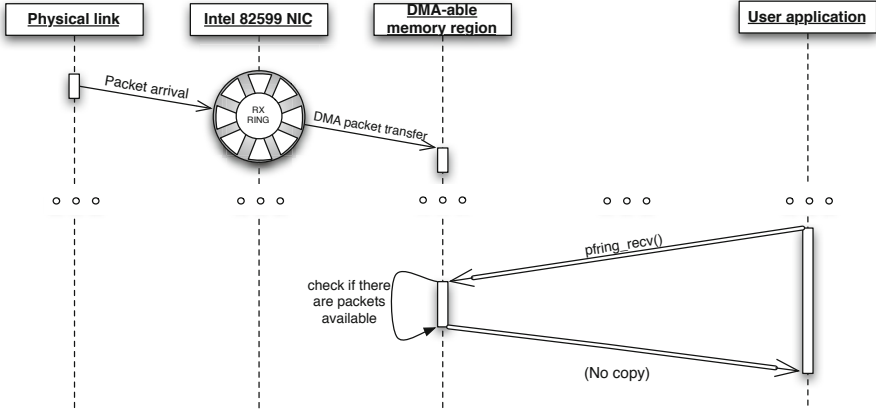


Fig. 7. PF_RING DMA RX scheme

the authors highlight the importance of NUMA-aware data placement in the performance of its engine. Similarly, it provides parallelism to packet processing at user-level, which balances CPU load and gives scalability in the number of cores and queues.

To reduce the per-packet processing overhead, batching techniques are utilized in user-level. For each batch request, the driver copies data from the huge packet buffer to a consecutive mapped memory region which is accessed from user-level. In order to eliminate the inherent cache misses, the modified device driver prefetches the next packet (both packet data and packet descriptor) while the current packet is being processed.

PS API works as follows: (i) user application opens a char device to communicate with the driver, `ps_init_handle()`, (ii) attaches to a given reception device (queue) with an `ioctl()`, `ps_attach_rx_device()`, and (iii) allocates and maps a memory region, between the kernel and user levels to exchange data with the driver, `ps_alloc_chunk()`. Then, when a user application requests for packets by means of an `ioctl()`, `ps_recv_chunk()`, PS driver copies a batch of them, if available, to the kernel packet buffer. PS interaction with users' applications during the reception process is summarized in Fig. 8.

PS I/O engine is available for the community¹³. Along with the modified Linux driver for Intel 82598/82599-based NICs network interface cards, a user library is released in order to ease the usage of the driver. The release also includes several sample applications, namely: a simplified version of `tcpdump`¹⁴, an `echo` application which sends back all traffic received by one interface, and a packet generator which is able to generate UDP packets with different 5-tuple combinations at maximum speed.

¹³ shader.kaist.edu/packetshader/io_engine/index.html

¹⁴ www.tcpdump.org

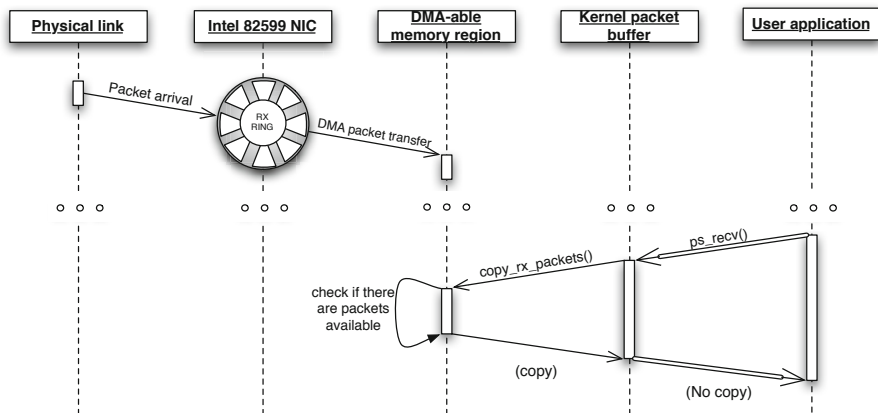


Fig. 8. PacketShader RX scheme

4.3 Netmap

Netmap proposal shares most of the characteristics of PacketShader’s architecture. That is, it applies memory pre-allocation during the initialization phase, buffers of fixed sizes (also of 2048 bytes), batch processing and parallel direct paths. It also implements memory mapping techniques to allow users’ application to access to kernel packet buffers (direct access to NIC is protected) with a simple and optimized metadata representation.

Such simple metadata is named netmap memory ring, and its structure contains information such as the ring size, a pointer to the current position of the buffer (*cur*), the number of received packets in the buffer or the number of empty slots in the buffer, in reception and transmission buffers respectively (*avail*), flags about the status, the memory offset of the packet buffer, and the array of metadata information; it has also one slot per packet which includes the length of the packet, the index in the packet buffer and some flags. Note that there is one netmap ring for each RSS queue, reception and transmission, which allows implementing parallel direct paths.

Netmap API usage is intuitive: first, a user process opens a netmap device with an `ioctl()`. To receive packets, users ask the system the number of available packets with another `ioctl()`, and then, the lengths and payloads of the packets are available for reading in the slots of the `netmap_ring`. This reading mode is able to process multiple packets in each operation. Note that netmap supports blocking mode through standard system calls, such as `poll()` or `select()`, passing the corresponding netmap file descriptors. In addition to this, netmap comes with a library that maps `libcap` functions into own netmap ones, which facilitates its operation. As a distinguish characteristic, Netmap works in an extensive set of hardware solutions: Intel 10 Gb/s adapters and several 1Gb/s adapters—Intel, RealTek and nVidia. Netmap presents other additional functionalities as, for example, packet forwarding.

Netmap framework is available for FreeBSD (HEAD, stable/9 and stable/8) and for Linux¹⁵. The current netmap version consists of 2000 lines for driver modifications and system calls, as well as a C header file of 200 lines to help developers to use netmap’s framework from user applications.

4.4 PFQ

PFQ is a novel packet capture engine that allows packet sniffing in user applications with a tunable degree of parallelism. The approach of PFQ is different from the previous studies. Instead of carrying out major modifications to the driver in order to skip the interrupt scheme of NAPI or map DMA-able memory and kernel packet buffers to user-space, PFQ is a general architecture that allows using both modified and vanilla drivers.

PFQ follows NAPI to fetch packets but implements two novel modifications once packets arrive at the kernel packet buffer with respect to the standard networking stack. First, PFQ uses an additional buffer (referred as batching queue) in which packets are copied once the kernel packet buffer is full, those packets are copied in a single batch that reduces concurrency and increases memory locality. This modification may be classified both as a batching and memory affinity technique. As a second modification, PFQ makes the most of the parallel paths technique at kernel level, that is, all its functionalities execute in parallel and in a distributed fashion across the system’s cores which has proven to minimize the overhead. In fact, PFQ is able to implement a new layer, named Packet Steering Block, in between user-level and batching queues, providing some interesting functionalities. Such layer distributes the traffic across different receive sockets (without limitation on the number of queues than can receive a given packet). These distribution tasks are carried out by means of memory mapping techniques to avoid additional copies between such sockets and the user level. The Packet Steering Block allows a capture thread to move a packet into several sockets, thus a socket may receive traffic from different capture threads. This functionality circumvents one of the drawbacks of using the parallel paths technique, that is, scenarios where packets of different flows or sessions must be analyzed by different applications—as explained in Section 3. Fig. 9 shows a temporal scheme of the process of requesting a packet in this engine.

It is worth remarking that, as stated before, PFQ obtains good performance with vanilla drivers, but using a patched driver with minimal modifications (a dozen lines of code) improves such performance. The driver change is to implement memory pre-allocation and re-use techniques.

PFQ is an open-source package which consists of a Linux kernel module and a user-level library written in C++¹⁶. PFQ API defines a `pfq` class which contains methods for initialization and packet reception. Whenever a user wants to capture traffic: (i) a `pfq` object must be created using the provided C++ constructor, (ii) devices must be added to the object calling its `add_device()` method, (iii)

¹⁵ info.iet.unipi.it/~luigi/netmap/

¹⁶ Available under GPL license in netserv.iet.unipi.it/software/pfq

timestamping must be enabled using `toggle_time_stamp()` method, and (iv) capturing must be enable using `enable()` method. After the initialization, each time a user wants to read a group of packets, the `read()` method is called. Using a custom C++ iterator provides by PFQ, the user can read each packet of the received group. When a user-level application finishes `pfq` object is destroyed by means of its defined C++ destructor. To get statistics about the received traffic `stats()` method can be called.

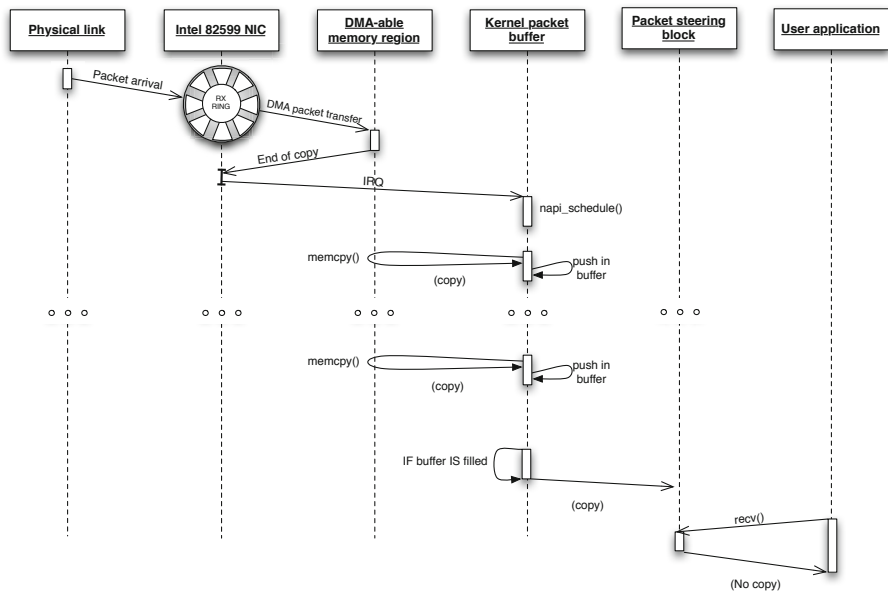


Fig. 9. PFQ RX scheme

4.5 Capture Engines Comparison and Discussion

Once we have detailed the main characteristics of the most prominent capture engines in the literature, we turned our focus to their performance in terms of percentage of packets correctly received. It is noteworthy that the comparison between them, from a quantitative standpoint, is not an easy task for two reasons: first, the hardware used by the different studies is not equivalent (in terms of type and frequency of CPU, amount and frequency of main memory, server architecture and number of network cards); second, the performance metrics used in the different studies are not the same, with differences in the type of traffic, and measurement of the burden of CPU or memory.

For such reason, we have stress tested the engines described in the previous section, on the same architecture. Specifically, our testbed setup consists of two machines (one for traffic generation purposes and another for receiving traffic and evaluation) directly connected through a 10 Gb/s fiber-based link. The receiver side is based on Intel Xeon with two processor of 6 cores each running at 2.30 GHz, with 96 GB of DDR3 RAM at 1333 MHz and equipped with a 10 GbE

Intel NIC based on 82599 chip. The server motherboard model is Supermicro X9DR3-F with two processor sockets and three PCIe 3.0 slots per processor, directly connected to each processor, following a similar scheme to that depicted in Fig. 2b. The NIC is connected to a slot corresponding to the first processor. The sender uses a HitechGlobal HTG-V5TXT-PCIe card which contains a Xilinx Virtex-5 FPGA (XC5VTX240) and four 10 GbE SFP+ ports. Using such a hardware-based sender guarantees accurate packet-interarrivals and 10 Gb/s throughput regardless of packet sizes.

We have taken into account two factors, the number of available queues/cores and packet sizes, and their influence into the percentage of correctly received packets. We assume a link of 10 Gb/s full-saturated with constant packet sizes. For example, 60-byte packets in a 10 Gb/s full-saturated link gives a throughput in Mpps of 14.88: $10^{10} / ((60 + 4 \text{ (CRC)} + 8 \text{ (Preamble)} + 12 \text{ (Inter-Frame Gap)}) \cdot 8)$. Equivalently, if packet sizes grow to 64 bytes, the throughput in Mpps decreases to 14.22, and so forth.

It is worth remarking that netmap does not appear in our comparison because its Linux version does not allow changing the number of receive queues being this fixed at the number of cores. As our testbed machine has 12 cores, in this scenario netmap capture engine requires allocating memory over the kernel limits, and netmap does not start. However, we note that according to [8], its performance figures should be similar to those from PacketShader. Regarding PFQ, we evaluated its performance installing the aware driver.

Before showing and discussing the performance evaluation results, let us describe the commands and applications used to configure the driver and receive traffic, for each capture engine. In the case of PF_RING, we installed driver using the provided script `load_dna_driver.sh`, changing the number of receive queues with the `RSS` parameter in the insertion of the driver module. To receive traffic using multiple queues, we executed the following command: `pfcount_multichannel -i dna0 -a -e 1 -g 0:1: . . . :n`, where `-i` indicates the device name, `-a` enables active waiting, `-e` sets reception mode and `-g` specifies the thread affinity for the different queues. Regarding PS, we installed the driver using the provided script `install.py` and receive packets using a slightly modified version of the provided application `echo`. Finally, in the case of PFQ, we install the driver using `n` reception queues, configure the receive interface, `eth0`, and set the IRQ affinity with the followings commands: `insmod ./ixgbe.ko RSS=n,n; ethtool -A eth0 autoneg off rx off tx off; bash ./set_irq_affinity.sh eth0`. To receive packets from `eth0` using `n` queues with the right CPU affinity, we ran: `./pfq-n-counters eth0:0:0 eth0:1:1 . . . eth0:n:n`. Note that in all cases, we have paid attention to NUMA affinity by executing the capture threads in the processor that the NIC is connected, as it has 6 cores, this is only possible when there are less than seven concurrent threads. In fact, ignoring NUMA affinity entails extremely significant performance cuts, specifically in the case of the smallest packet sizes, this may reduce performance by its half.

First, Fig. 10 aims at showing the worst case scenario of a full-saturated 10 Gb/s link with packets of constant size of 60 bytes (as in the following, excluding

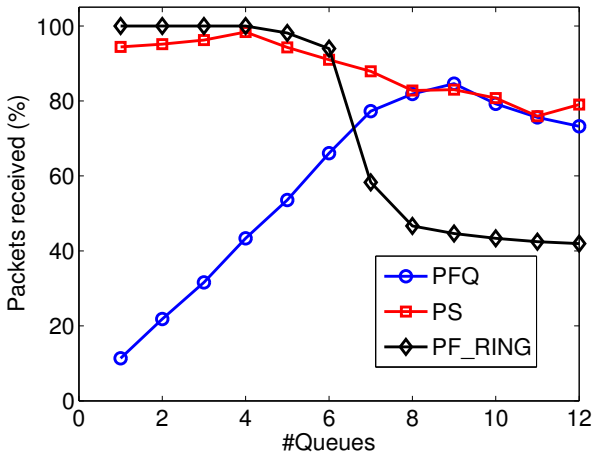


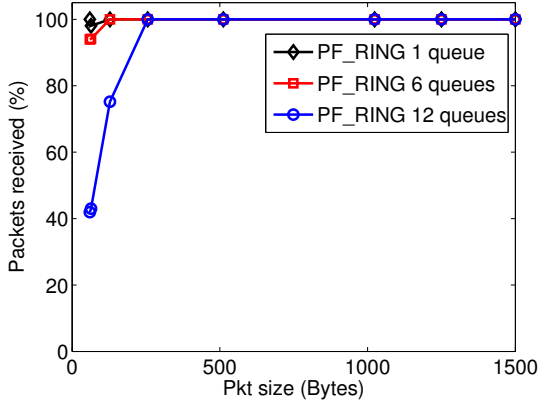
Fig. 10. Engines’ performance for 60 (+4 CRC) byte packets

Ethernet CRC) for different number of queues (ranging from 1 to 12). Note that this represents a extremely demanding scenario, 14.88 Mpps, but probably not very realistic given that the average Internet packet size is clearly larger [22].

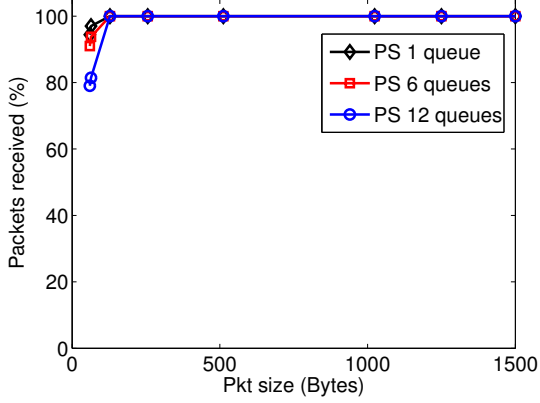
In this scenario, PacketShader is able to handle nearly the total throughput when the number of queues ranges between 1 and 4, being with this latter figure when the performance peaks. Such relatively counterintuitive behavior is shared by PF_RING DNA system, which shows its best permanence, a remarkable 100% packet received rate, with a few queues, whereas with when number of queues is larger than 7, the performance dips. Conversely to such behavior, PFQ increases its performance according to the number of queues up to its maximum with nine queues, when such growth stalls. To further investigate such phenomenon, Fig. 11 depicts the results for different packets sizes (60, 64, 128, 256, 512, 1024, 1250 and 1500 bytes) and one, six and twelve queues.

PF_RING DNA shows the best results with one and six queues. It does not show packet losses for all scenarios but those with packet sizes of 64 bytes and, even in this case, such figure is very low (about 4% with six queues and lower than 0.5% with one). Surprisingly, increasing packet sizes from 60 to 64 bytes, entails a degradation in the PF_RING DNA performance, although beyond that packe size, the performance recovers 100% rates. Note that larger packet sizes implies directly lower throughputs in Mpps. According to [8], investigation in this regard has shown that this behavior is because of the design of NICs and I/O bridges that make certain packet sizes to fit better with their architectures.

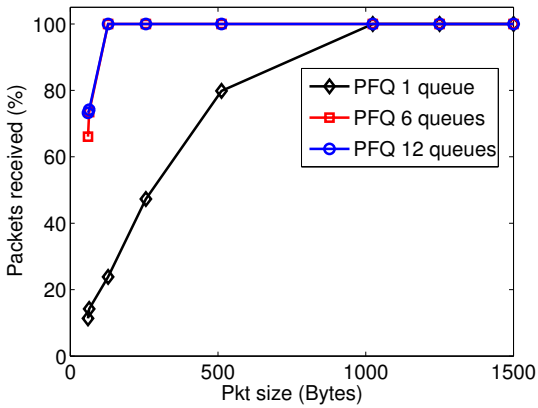
In a scenario in which one single user-level application is unable to handle all the received traffic, may result of interest to use more than one receive queue (with one user-level application per queue). In our testbed and assuming twelve queues, PacketShader has shown comparatively the best result, although, as PF_RING DNA, it performs better with a fewer number of queues. Specifically, for packet sizes of 128 bytes and larger ones, it achieves full packet received rates,



(a) PF_RING DNA



(b) PacketShader



(c) PFQ

Fig. 11. Performance in terms of percentage of received packet for different number of queues and constant packet sizes for a full-saturated 10 Gb/s link

regardless the number of queues. With the smallest packets sizes, it gives loss ratios of 20% in its worst case of twelve queues, 7% with six, and about 4% with one queue.

Analyzing PFQ's results, we note that such engine achieves also 100% received packet rates, but conversely to the other approaches, it works better with several queues. It requires at least six ones to achieve no losses with packets of 128 bytes or more, whereas with one queue, packets must be larger or equal to 1000 bytes to achieve full rates. This behavior was expected due to the importance of parallelism in the implementation of PFQ.

We find that these engines may cover different scenarios, even the more demanding ones. We state two types of them, whether we may assume the availability of multiple cores or not, and whether the traffic intensity (in Mpps) is extremely high or not (for example, packet size averages smaller than 128 bytes, which is not very common). That is, if the number of queues is not relevant, given that the capture machine has many available cores or no other process is executing but the capture process itself, and the intensity is relatively low (namely, some 8 Mpps), PFQ seems to be a suitable option. It comprises a socket-like API which is intuitive to use as well as other interesting functionalities, such as an intermediate layer to aggregate traffic, while it achieves full received packet rates for twelve queues. On the other hand, if traffic intensity is higher than the previous assumption, PacketShader presents a good compromise between the number of queues and performance.

Nonetheless, often multi-queue scenarios are not adequate. For example, accurate timestamps may be necessary [19], packet disorder may be a significant inconvenient (according to the application running on the top of the engine) [18], or simply, it may be interesting to save cores for other tasks. In this scenario, PF_RING DNA is clearly the best option, as it shows (almost) full rates regardless packet sizes even with only one queue (thus, avoiding any drawbacks due to parallel paths).

5 Conclusions

The utilization of commodity hardware in high-performance tasks, previously reserved to specialized hardware, has raised great expectation in the Internet community, given the astonishing results that some approaches have attained at low cost. In this chapter we have first identified the limitations of the default networking stack and shown the proposed solutions to circumvent such limitations. In general, the keys to achieve high performance are efficient memory management, low-level hardware interaction and programming optimization. Unfortunately, this has transformed network monitoring into a non trivial process composed of a set of sub-tasks, each of which presents complicated configuration details. The adequate tuning of such configuration has proven of paramount importance given its strong impact on the overall performance. In this light, this chapter has carefully reviewed and highlighted such significant details, providing practitioners and researchers with a road-map to implement high-performance

networking systems in commodity hardware. Additionally, we note that this effort of reviewing limitation and bottlenecks and their respective solutions may be also useful for other areas of research and not only for monitoring purposes or packet processing (for example, virtualization).

This chapter has also reviewed and compared successful implementations of packet capture engines. We have identified the solutions that each engine implements as well as their pros and cons. Specifically, we have found that each engine may be more adequate for a different scenario according to the required throughput and availability of processing cores in the system. As a conclusion, the performance results exhibited in this chapter, in addition to the inherent flexibility and low cost of the systems based on commodity hardware, make this solution a promising technology at the present.

Finally, we highlight that the analysis and development of software based on multi-core hardware is still an open issue. Problems such as the aggregation of related flows, accurate packet timestamping, and packet disordering will for sure receive more attention by the research community in the future.

References

1. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review* 38(2), 69–74 (2008)
2. Braun, L., Didebulidze, A., Kammenhuber, N., Carle, G.: Comparing and improving current packet capturing solutions based on commodity hardware. In: *Proceedings of ACM Internet Measurement Conference* (2010)
3. Dobrescu, M., Egi, N., Argyraki, K., Chun, B.G., Fall, K., Iannaccone, G., Knies, A., Manesh, M., Ratnasamy, S.: Routebricks: exploiting parallelism to scale software routers. In: *Proceedings of ACM SIGOPS Symposium on Operating Systems Principles* (2009)
4. Han, S., Jang, K., Park, K.S., Moon, S.: PacketShader: a GPU-accelerated software router. *ACM SIGCOMM Computer Communication Review* 40(4), 195–206 (2010)
5. Mogul, J., Ramakrishnan, K.K.: Eliminating receive livelock in an interrupt-driven kernel. *ACM Transactions on Computer Systems* 15(3), 217–252 (1997)
6. Kim, I., Moon, J., Yeom, H.Y.: Timer-based interrupt mitigation for high performance packet processing. In: *Proceedings of the Conference on High Performance Computing in the Asia-Pacific Region* (2001)
7. Fusco, F., Deri, L.: High speed network traffic analysis with commodity multi-core systems. In: *Proceedings of ACM Internet Measurement Conference* (2010)
8. Rizzo, L.: Netmap: a novel framework for fast packet I/O. In: *Proceedings of USENIX Annual Technical Conference* (2012)
9. Bonelli, N., Di Pietro, A., Giordano, S., Procissi, G.: On Multi-gigabit Packet Capturing with Multi-core Commodity Hardware. In: Taft, N., Ricciato, F. (eds.) *PAM 2012*. LNCS, vol. 7192, pp. 64–73. Springer, Heidelberg (2012)
10. Rizzo, L., Deri, L., Cardigliano, A.: 10 Gbit/s line rate packet processing using commodity hardware: survey and new proposals (2012), <http://luca.ntop.org/10g.pdf>
11. Intel: 82599 10 Gbe controller datasheet (2012), <http://www.intel.com/content/www/us/en/ethernet-controllers/82599-10-gbe-controller-datasheet.html>

12. Microsoft: Receive Side Scaling, [http://msdn.microsoft.com/en-us/library/windows/hardware/ff567236\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/hardware/ff567236(v=vs.85).aspx)
13. Woo, S., Park, K.: Scalable TCP session monitoring with Symmetric Receive-Side Scaling. Technical report KAIST (2012), <http://www.ndsl.kaist.edu/~shinae/papers/TR-symRSS.pdf>
14. Dobrescu, M., Argyraki, K., Ratnasamy, S.: Toward predictable performance in software packet-processing platforms. In: Proceedings of USENIX Symposium on Networked Systems Design and Implementation (2012)
15. Zabala, L., Ferro, A., Pineda, A.: Modelling packet capturing in a traffic monitoring system based on Linux. In: Proceedings of Performance Evaluation of Computer and Telecommunication Systems (2012)
16. Liao, G., Znu, X., Bnuyan, L.: A new server I/O architecture for high speed networks. In: Proceedings of Symposium on High-Performance Computer Architecture (2011)
17. Papadogiannakis, A., Vasiliadis, G., Antoniadis, D., Polychronakis, M., Markatos, E.P.: Improving the performance of passive network monitoring applications with memory locality enhancements. *Computer Communications* 35(1), 129–140 (2012)
18. Wu, W., DeMar, P., Crawford, M.: Why can some advanced Ethernet NICs cause packet reordering? *IEEE Communications Letters* 15(2), 253–255 (2011)
19. Moreno, V., Santiago del Río, P.M., Ramos, J., Garnica, J.J., García-Dorado, J.L.: Batch to the future: Analyzing timestamp accuracy of high-performance packet I/O engines. *IEEE Communications Letters* 16(11), 1888–1891 (2012)
20. Su, W., Zhang, L., Tang, D., Gao, X.: Using direct cache access combined with integrated NIC architecture to accelerate network processing. In: Proceedings of IEEE Conference on High Performance Computing and IEEE Conference on Embedded Software and Systems (2012)
21. Krasnyansky, M.: UIO-IXGBE (2012), <https://opensource.qualcomm.com/wiki/UIO-IXGBE>
22. CAIDA: Traffic analysis research (2002-2012), http://www.caida.org/data/passive/trace_stats/

Active Techniques for Available Bandwidth Estimation: Comparison and Application

Alessio Botta¹, Alan Davy², Brian Meskill², and Giuseppe Aceto¹

¹ University of Napoli Federico II, Italy

{a.botta,giuseppe.aceto}@unina.it

² Waterford Institute of Technology, Ireland

{adavy,bmeskill}@tssg.org

Abstract. There are various parameters for analyzing the quality of network communication links and paths, one attracting particular attention is available bandwidth. In this chapter we describe a platform for the available bandwidth estimation, a comparison of different tools for the estimation of this parameter, and an application of such estimation in a real-world application. In details, we describe a novel platform called UANM, capable of properly choosing, configuring, and using different available bandwidth tools and techniques in an autonomic fashion. Moreover, thanks to UANM, we show the results of a comparison of the performance of several tools in terms of accuracy, probing time and intrusiveness. Finally, we show a practical example of the use of the available bandwidth measurement: we describe an approach for server selection and admission control in a content distribution network based on the available bandwidth estimation.

Keywords: Available Bandwidth, Quality of Service, Server Selection.

1 Introduction

In recent years, there has been an increased focus on measuring the quality of communication links over data networks [9,8]. These communication links can represent logical or physical connections between two entities at various layers in the protocol stack, and over varying heterogeneous networking technologies. With the rise of application level traffic optimization within various application domains such as multimedia streaming [23] and content delivery server selection [21,22], the assessment of the available bandwidth of communication links is invaluable. This chapter focuses on the estimation of the available bandwidth, a platform used for comparison between various available bandwidth estimation tools, and finally a usage scenario focusing on server selection. When looking beyond a single link and at the complete end-to-end path of a communication network, the measurement of available bandwidth becomes imperative [5]. This is generally termed the residual bandwidth available on the end-to-end path before additional congestion occurs. This measurement is useful for a variety of applications such as server selection, peer node selection, and video streaming.

There is a wide set of tools available for estimating this metric, each tailored to specific scenarios and requiring expert operators to obtain accurate results. In this paper we describe a novel platform called UANM, which has been designed to avoid the effect of the interference among concurrent measurement processes, and to automatically select and configure a measurement technique, according to the scenario. Moreover, offering a common generic model for the measurement tools (the *plugin API*) UANM allows for a fair comparison of the different techniques in terms of probing time, intrusiveness, and accuracy in each given scenario. In this paper, we show the results of a comparative analysis of different available bandwidth tools and techniques performed through UANM under varying network scenarios. Moreover, to show a use-case scenario for the Available Bandwidth estimation, we present an admission control and server selection framework. This framework relies on the measurement of available bandwidth between the client and the server to both choose a suitable server to deliver content and also assess whether adequate bandwidth is available to serve the request. We demonstrate that with appropriately configured available bandwidth measurements, improved control of end-to-end traffic flows over an unmanaged network infrastructure is possible.

The chapter is structured as follows: Section 2 provides background details and related work on the concepts of available bandwidth. Section 3 discusses the UANM platform and a comparison of available bandwidth estimation tools on the UANM platform. Section 4 demonstrates the application of available bandwidth estimation to admission control of video content over an unmanaged network. Finally, Section 5 concludes the chapter with a summary and future research challenges in the area of quality analysis of communication links.

2 Related Work

Available Bandwidth of a network path is defined as its remaining capacity, that is, the amount of traffic that can be sent along the path without congesting it [10]. Many tools exist for calculating it in an end-to-end context; that is, without any information from the intermediate topology. Such tools vary in the model describing the network traffic and in the estimation technique, thus varying also in accuracy and other properties of practical interest. The many tools that have been proposed in the literature are often broadly categorized into two approaches, which we briefly discuss via some well known examples.

The Probe Gap Model (PGM)[17] approach uses probe packet pairs or packet trains to determine the available bandwidth. It uses these pairs by noting the difference between their network entry time gap, and their network exit time gap. The difference in this time gap is the time the bottleneck link required to service any non probing traffic on the bottleneck hop and this time can be used, along with the link capacity, to calculate the available bandwidth of the bottleneck link[34]. The Probe Rate Model[14] approach sends a train of packets and utilizes the concept of self-induced congestion[35] to determine the available bandwidth of the network path. Each packet train is forwarded through the network at a

particular rate. The rate increases until particular self-induced characteristics are observed from the packet train such as a diversion from the initial packet train transmission rate. This method can pinpoint the available bandwidth of the path without knowledge of the physical capacity of the network links involved. Each of these approaches has its pros and cons, and none of them is better than the others in all the possible application scenarios. For this reason, the scientific community needs tools and platforms for using what is already available at the best of its possibilities, and, to this aim, several research papers in literature have compared the performance of available bandwidth estimation tools. Most of the comparison works have been done when presenting a new available bandwidth estimation tool. For instance, when presenting Traceband [16], the authors compared its performance with that of Spruce and Pathload on a real network using different traffic patterns. Results show that Traceband is faster and less intrusive than the others, and it achieves the same accuracy of Pathload. On the other hand, there are also works in which the authors compared several well known tools, without presenting new ones. For example, Goldoni et al. [15] compared the accuracy, the intrusiveness, and the convergence time of nine of the most widespread tools on a real testbed with 100Mbps links, and with both constant bit rate (CBR) and Poisson cross-traffic. Other works [31,24] evaluated such performance on very high speed networks. Some works [5] have also shown that the combined use of different techniques can increase the estimation accuracy. Summarizing, a first issue to be tackled for a proper available bandwidth estimation is the choice of the right tool according to the scenario.

Another important issue for the available bandwidth estimation is the fact that most of the existing tools can provide accurate results only if properly used and calibrated. This issue has been revealed and analyzed by different works in literature. For example, in 2004 Paxson et al. [27] claimed that a better design stage of measurement experiments is of great importance to avoid frequent mistakes, mainly due to the imperfections of tools. The authors basically reported that a calibration is usually needed to detect and correct possible errors. Other works [3,34] analyzed commonly used tools on real testbeds and reported several pitfalls in which they can typically end.

These two issues are among the main motivations that drove us to design and implement UANM, a platform that helps the user to obtain accurate, fast and non intrusive available bandwidth estimations, choosing and configuring the right tool for the operating scenario. Architectures similar to UANM, which aim at taking all the measurement-related variables into the proper account, have already been presented in literature, although not specifically designed for the available bandwidth measurement. NetQuest [33], for example, firstly designs the experiments in order to best fit to the current scenario, and then it builds a global view of the network status. However, differently from UANM, NetQuest is not mainly interested in obtaining the best performance from the available bandwidth estimation, since it is more oriented to a wider knowledge of the general network status. Wide-scale infrastructures for network measurement and experimentation such as GENI [13] and Planetlab [12] have been created

in order to perform experiments on a global level and study Internet-scale phenomena. Being designed to interact with third party measurement tools, UANM can leverage the existing infrastructures to perform the requested measurements; moreover, thanks to its *user API*, it can be easily integrated in such infrastructures as a (compound) measurement tool. Similarly to our approach, Sommers et al. [32] proposed YAZ, an architecture whose main goal is to calibrate the existing tools in order to obtain the best results from the measurements. Differently from UANM, YAZ does not support concurrent experiments, does not consider the network status, and does not support third-party tools.

3 Comparing Available Bandwidth Estimation Tools through UANM

UANM is a distributed platform for network measurement supporting different techniques and tools. Full compliance and open interaction with existing tools is retained while offering a fair comparison environment, mutual exclusion of concurrent measurements and automatic selection and calibration of the tools. We report a brief description of UANM. More details are reported in [2,1].

3.1 UANM Architecture

The components of the platform are of four types: daemons, clients, measurement plugins and third-party probes. The daemons are in charge of the orchestration of the measurements requested by the clients, by managing the plugins and interacting with the other daemons; the actual measurements are performed by means of the daemon-managed plugins cooperating with other plugins or third-party probes. UANM considers the plugins as gray-boxes, using only high level methods such as `initializePlugin()`, `startMeasure()`, ignoring underlying details. This eases the transformation of third-party tools in UANM-plugins with minimal changes to the original code and retains full compatibility of the UANM-plugin version and the original standalone version of the tool. Moreover, this allows UANM to be not tied to a single type of measurement or technique. Currently, we are focusing on available bandwidth measurement because we believe that this research area can particularly benefit from this platform. Therefore, the following plugins have been currently implemented: Abing [25], Assolo [14], Diettopp [20], IGI [17], Pathchirp [29], Pathload [19], Spruce [34], Wbest [23]. The interactions between clients and daemons follow a client-server paradigm, while inter-daemons communications happen on peer-to-peer basis; both types of communication use a dedicated control protocol. An API in C allows external applications and platforms to act as clients. The communications between measurement plugins and with third-party probes use the plugin-specific control protocol. A diagram of the possible interactions among the components of the platform is reported in Fig.1. To illustrate how UANM works, in the following we describe the typical sequence of actions for a measurement experiment and the components involved.

- The client issues a measurement request to a daemon that runs on one of the edges of the path under test. The request can specify measurement constraints according to the intended purpose of the measure (e.g. server selection as described in Sec. 4). As an example, to perform server selection an application may request quick-and-dirty estimations towards the possible servers, while for continuous network monitoring a series of non intrusive measurements could be desired. Detailed constraints can be specified such as the averaging timescale, the number of subsequent estimations, the total probe load, or even the specific measurement technique and its parameters.
- The daemon performs a feasibility check to assess whether the request can be fulfilled and, in case the client did not request a specific one, which plugin is best fitted to the current *context*. The *context* comprises a description of the optional measurement constraints as well as the information about the measurement process, both structural (e.g. wireless hops or broadband access links along the path, UANM instances or known third-party estimation tools on the other edge, etc.) and behavioral (e.g. congested path, rapidly changing routes, etc.). This phase may imply communications with the other instrumented edge in order to update the *context*. If no constraints have been specified, the measurement will be set up to reach a trade-off between accuracy and intrusiveness. A daemon module called *decision engine* uses this information to select and configure the suitable plugin.
- A daemon module called *scheduler* schedules the measurement according to the policies (currently FCFS). This phase is needed because active estimation tools can be unreliable in case of concurrent measurements that involve shared resources (see Sec.3.2 for further details).
- The actual measurement process is performed executing the measurement phase of the plugin, in mutual exclusion with other possible measurements scheduled on the same path or on the same daemon.
- The result of the measurement is returned to the client, and it is also used to update the known *context*, allowing for better future decisions.

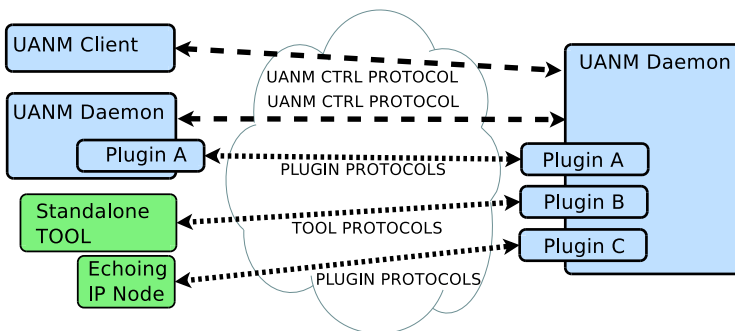


Fig. 1. UANM components and communications

UANM is geared towards wide adoption among application developers and researchers in the field of network measurement. A prototype is released under the GPL terms, with a LGPL API for the development of measurement plugins. We believe that UANM allows to overcome the main limitations of current tools, providing accurate available bandwidth estimation in heterogeneous environments.

3.2 UANM Highlights

UANM has been conceived with two main objectives: (i) the avoidance of the effect of the interference among concurrent measurement processes, and (ii) the automatic selection of a tool and its parameters in order to increase the performance. Moreover, offering a common generic model for the measurement tools (the *plugin API*) allows for a fair comparison of the different techniques in terms of probing time, intrusiveness, and accuracy in each given scenario. As for the first point, the problem a measure may encounter when more uncontrolled measurement processes share even one single part of the network for a long or short time interval has been shown by different works in literature [4,5,6,2,1]. This problem may occur since the current methods and tools do not provide coordination among measurement stations or any kind of alert feedback from the network.

In our former paper [1] we have shown the results of experiments aimed at assessing the capability of UANM in avoiding interference from concurrent measurements. In such experiments, using some of the most utilized available bandwidth estimation tools alone and concurrently, we experimentally verified and quantified the interference effect: in the concurrent case, the error is up to 3.5 times the error in the stand alone one. We also noticed that, in some experiments, the tools (Pathload in particular) did not converge to a final stage if run concurrently. This result has been attributed to the approach adopted by this tool, which leads the network towards congestion. Finally, we could assess that the design of UANM daemon avoids the interference effect thanks to the scheduler that coordinates the different clients. We refer the interested readers to [1] for more details.

The second aspect we highlight is the lack of measurement accuracy that may happen when the available bandwidth tools are used without knowledge of the basic network configuration. In fact, currently available tools are designed or tuned by default for a given scenario (layer 1 technology, intrusiveness of the measurement traffic, desired accuracy, maximum measurement time), and therefore are hardly suitable for as-is integration in third party software and automated use, as the technical knowledge of the tool and of the possibly varying scenario could be missing. For instance, Pathchirp gives wrong results on high speed networks unless some of its parameters are correctly set up. The decision engine in UANM is able to automatically select and configure the best tool for this scenario. In pursuing the second objective the design of UANM follows the autonomic paradigm [18] as the interface presented to the user can be characterized as *Sensor* (providing the results of the measurements) and *Effector*, allowing the user to specify the *policies*, not the execution details of the measurements (though the latter possibility is offered as a special case for the experimenter).

The outcome of the request issued by the user is the result of a continuously running sequence that follows the autonomic control flow, presenting

- a *knowledge base* containing the current context (the available measurement tools and their characteristics, as well as the available information about the network)
- a *planning* phase (managing the scheduling of possible concurrent measurements)
- the *execution* of the scheduled plan (activation of the measurement tools)
- the *monitoring* of the results of the measurements and their *analysis* (that update the context and offer a data report to the user that requested it).

To evaluate experimentally the efficacy of the automatic management of the measurement tools, we compared the accuracy achievable with Pathchirp or Pathload with that achievable when these tools are used through UANM on 1Gbps Ethernet links. The results showed that Pathchirp and Pathload are quite inaccurate (with relative errors up to 90%) with the former being the least accurate. In this scenario UANM selected Pathchirp with a different configuration of the packet trains, obtaining the highest accuracy in all the tested load conditions (with relative error 19% where the standalone configuration had 90%). We refer the interested readers to [1] for more details.

3.3 UANM: Comparing Available Bandwidth Estimation Tools

In this section we present the results of a fair comparison of the performance of several available bandwidth estimation tools, performed by means of UANM.

Testbed and Tools. These experiments have been conducted on the laboratory testbed depicted in Fig. 2. The end hosts are provided with *Pathchirp* and *Pathload*, as well as with UANM (equipped with the eight plugins reported in Section 3.1). On the end hosts, we also installed a traffic generator called D-ITG [7] to generate the cross traffic in order to reproduce different network load conditions. Different hosts are used for cross- and probe-traffic generator in order to avoid interference between these activities.

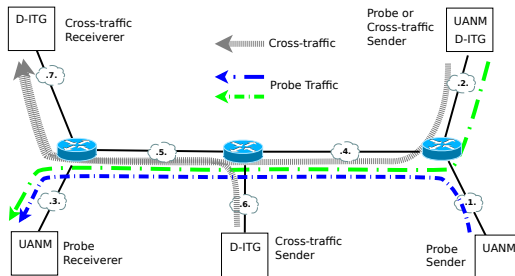


Fig. 2. Testbed used for the comparison of the performance of the tools

As we wanted to compare the techniques implemented by the tools in their standard operating conditions, we run all the plugins with default values of the configuration parameters. For the same reason, no specific settings have been enforced for the operating systems of the testbed nodes, and no change have been made on standard settings for the network adapters.

All the links in the testbed run at 100 Mbps, full duplex. CBR cross traffic has been generated by the host *Cross-traffic Sender* (bottom, center in the testbed) towards the host *Cross-traffic Receiver* (top, left in the testbed) at rates of 25, 50, 75 and 100 Mbps, and with IP packet size of 1000 Bytes. 10 measurements have been performed with each plugin and for each cross traffic rate, as well as with no cross traffic. The results shown in the following represent the average values of the 10 measures collected.

Results. In Fig. 3a we report the results obtained by the tools in terms of accuracy. As we can see, Abing, Wbest and Spruce achieved the worst performance. A recent work in literature [15] reported slightly different results. A deeper investigation revealed that the difference is due to the fact the whole testbed is equipped with Gigabit Ethernet network adapters, whose interrupt mitigation feature is known to affect the performance of some available bandwidth estimation tools [32,28]. Besides that, other important observations can be done: i) Wbest default parameters are suited to IEEE 802.11 wireless networks, therefore, its performance may be impacted by a full wired scenario; ii) Diettopp obtained the highest accuracy and also the smallest standard deviation; iii) Pathload also obtained accurate results, but the standard deviation is large, especially for small volumes of cross traffic; iv) Pathchirp achieved intermediate performance, both in terms of relative error and standard deviation; v) IGI/PTR obtained results similar to those of Pathload, and a small standard deviation.

Figure 3b shows the *probing time* of the tools, calculated as the difference between the timestamps of the last and the first probe packet: we consider this value as it represents both the time during which the network is solicited with additional traffic (for the calculation of intrusiveness), and the actual measurement interval (for the calculation of time averages and sampling rates); the time needed for the setup of the control channel and the mutual exclusion are not included in this value. In this figure, we can observe that the *probing time* of the tools is almost constant with the cross traffic rate, with the exceptions of IGI/PTR and Pathload. The probing time of IGI/PTR is increasing with the cross traffic: from 74ms with unloaded path, to 0.7s with 75 Mbps of cross traffic, and up to to 36 s with fully saturated path. Pathload obtained about the same probing times (i.e. 6 s) in almost all the load conditions, except at 100 Mbps, where it required about 27 s. The probing times of Assolo, Diettopp, Pathchirp and Spruce are comparable with those of Pathload ($\in [6, 8]$ s) with very low standard deviation.

Fig. 3c shows the volume of probe traffic generated by the tools for a measurement (in average). We notice that, besides IGI/PTR and Pathload, the volume

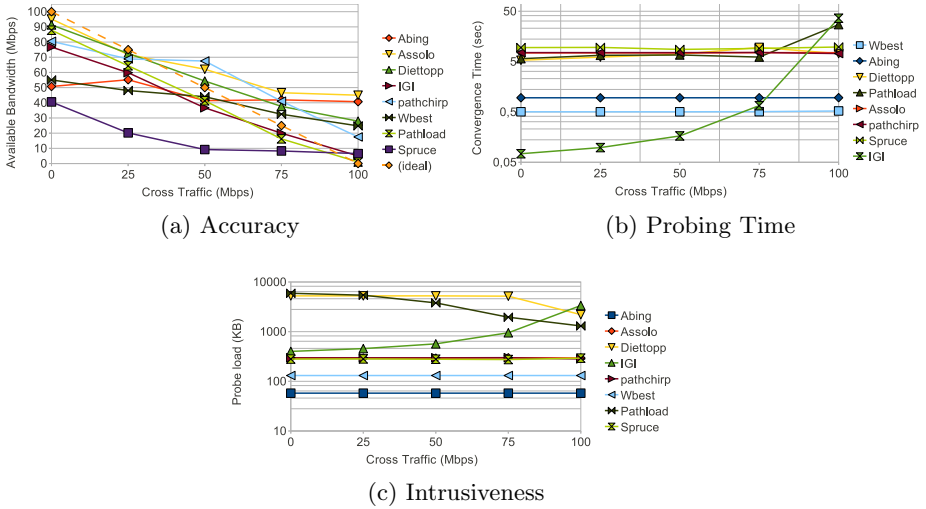


Fig. 3. A fair comparison among eight available bandwidth estimation tools in terms of accuracy, probing time (between first and last probing packet) and intrusiveness (total *volume of probe traffic* injected). The standard deviation is not visible in the plots. Its figures are reported in the text for interesting cases.

is almost independent of the cross traffic rate. The intrusiveness of IGI/PTR increases with the cross traffic volume: from 410 KB to 3.4 MB, and the standard deviation increased from 46 KB to 980 KB. Pathload has a probe-traffic volume decreasing with the cross traffic rate: from 6 MB down to 1.3 MB, and the standard deviation decreased from 2.8 MB down to 208 KB. This behavior is due to the way the technique works. Having to congest the path, Pathload generates a larger volume of probe traffic when the network is unloaded. Pathload and Diettopp are the most intrusive tool, while Abing is the least intrusive one.

Thanks to this fair comparison, a few considerations can be done. The most accurate tools are Diettopp and Pathload, both being also the most intrusive and the slowest (i.e. highest probing time). IGI/PTR is the fastest in all the cases except with the fully saturated path, and it has a good accuracy. But, its probing time can range in more than 2 orders of magnitude. Pathchirp, while being less accurate for low cross-traffic volumes, has a probing time comparable with Pathload and Diettopp, but it is also less intrusive. Assolo has performance similar to Pathchirp, but higher accuracy for smaller volumes of cross traffic. Even if we used only the default settings of the plugins and a basic network scenario (varying only the cross traffic rate), the experimental results showed that there is no tool that suits the requirements of all the applications. Therefore, an informed choice of the tool and of its parameters is necessary for the effective and efficient estimation of the available bandwidth. UANM has been conceived with this idea in mind, and it can be profitably be used for this aim.

4 Application of Available Bandwidth Estimation

In the case of content distribution networks supporting IPTV, Video on Demand (VoD) or similar applications, there is a necessity to adhere to Quality of Service (QoS) targets. To this end, admission control is an important component to the deployment of a successful IPTV/VoD solutions. In order to show a possible application of the estimation of the available bandwidth, this section discusses how this parameter can be used in an admission control framework [21]. Such a framework operates in an end-to-end manner without the need to access measurements directly from the network topology connecting end-user points of attachment to those of one or more content servers. In the presented framework Quality of Service is maintained by performing admission control based on the estimated current available bandwidth of the network paths.

4.1 IPTV Admission Control Using Available Bandwidth Estimates

The IPTV admission control framework is concerned with ensuring the adequate delivery of content from the content servers to the clients over an intermediate topology that is not controlled by the service provider (e.g. the Internet). Content servers have two purposes: to serve content items and to measure the available bandwidth between the content server and the destination edge router of the network. The edge router is also the point of attachment of the clients to the network. One server in the framework, referred to as the selection server, has the sole responsibility for collating the available bandwidth estimates for each path and making the decision on whether or not to accept a new request. This server works in conjunction with any number of content servers in the framework. For simplicity we assume that each content server contains a complete library of all the content items, allowing any item to be served from any server. An estimation of the available bandwidth between the content server and the edge router is performed at regularly defined intervals. The results are continuously sent to the selection server. Independently of this, when a request is generated, it goes from the client to the selection server. The selection server makes a decision to accept or reject the request (based on the algorithms discussed in the following sections). A rejection is reported to the client, whereas an acceptance is delegated to the appropriate content server and the content is served.

A Video on Demand content library can be expected to contain in the order of hundreds or thousands of different content items. The algorithm presented here places each item into categories and operates by dealing with these categories. This allows the algorithm to remain efficient whilst still dealing with a large scale content library as the amount of categories would be expected to be in the order of tens for even a very large content library. Items can be categorized into groups with similar durations, bandwidth requirements, and revenue potential or a subset of these characteristics. Two content items that have a similar duration and peak throughput might be placed into different categories due to one being a newer release and therefore having a higher earning potential. Similarly, two items might be categorized differently despite having the same revenue potential

if the durations are different enough to vary the cost involved in serving each content item.

We examine the performance of this framework under varying background traffic conditions within the network. We assume each content server has the resources required to serve any content assigned to it and that all flows assigned to a server run to completion. This allows us to focus on the performance of the admission control framework and not the processing capabilities of the content servers.

4.2 Available Bandwidth Based Server Selection / Admission Control Algorithm

A simple selection/admission control algorithm bases its decision to accept or reject a new request for a content item type on whether any of the content servers have the available bandwidth required on their path to the client. Assume there are I individual types of content made available by the service provider. Let $i = 1, \dots, I$ denote an arbitrary type of content item. Let $p(i)$ denote the peak bandwidth per second required by item type i . Assume that the service provider maintains J content servers, each with a single dedicated egress link to the core network. Let $j = 1, \dots, J$ denote an arbitrary content server. Let $\hat{B}_{jd}(t)$ denote the estimate of available bandwidth between content server j and edge router d as calculated at time t .

As mentioned previously, the selection server maintains an estimate of the Available Bandwidth $\hat{B}_{jd}(t)$ for the current time t of each path between content server j and edge router d . This estimate is calculated as a moving average of recent reported estimates. If only one server is listed as possessing enough bandwidth to support a request for a particular item type, then the request is accepted and allocated to that content server j^* . If there are multiple servers capable of supporting the request, j^* is assigned to be the content server with the highest available bandwidth. The final case occurs when none of the network paths have sufficient bandwidth and in this case the request is rejected. Once accepted to a server, a traffic flow will use this server for the duration of the flow. This is specified formally in Alg. 1.

4.3 Simulations and Results

The framework has been firstly implemented in simulation to carefully test its performance and evaluate the impact of different parameters before the deployment in real scenarios. The simulations were performed using the OPNET ModelerTM[26] simulation environment. The framework is deployed in a scenario where there is three different content servers (A,B,C). We use traffic traces taken from actual videos using various CODECs by [30] and use [11] to inform the distribution of mean durations, enabling us to create realistic traffic flows. To ensure a conservative use of available bandwidth at each server, a threshold of 90% is used at each server as an upper bound. This is to cater for multiple flows reaching peak throughput simultaneously. For the purposes of control and analysis,


```

Input:  $i^*, \{\hat{B}_{j_d}(t)\}$ 

forall the content servers  $j = 1 \dots J$  do
  List all content servers  $\{j'\}$  for which  $\hat{B}_{j_d}(t) > p(i^*)$ ;
  if  $\{j'\} \neq NULL$  then
    Select  $j^* \in \{j'\} : \hat{B}_{j^*_d} = \max\{\hat{B}_{j'_d}(t)\}$ ;
    return ACCEPT,  $j^*$ ;
  end
  else
    return REJECT;
  end
end

```

Algorithm 1. Available bandwidth admission control algorithm (ABAC)

we concentrate on requests arriving into the network from a single access point and we specify the intermediate topology between the content servers and the access point to contain Fast Ethernet (100Mbps) links.

For the estimation of the Available Bandwidth in this scenario we have to consider the specific (often contrasting) requirements. Firstly, higher probing rates lead to more accurate results, but the result takes longer to be generated and a heavier footprint is imposed on the network as the tool will inject more data into the path. Secondly, the available bandwidth estimation frequency, or, conversely, the inter-estimate time, can influence how long it takes for the estimate to update after a change in the bandwidth that is available: the shorter the time between estimates, the faster a tool can become aware of changes in the available bandwidth, but also the more intrusive is the probing. The application implementing the Server Selection / Admission Control algorithm would request UANM for a measurement profile characterized by relaxed constraints for probing time and accuracy, low intrusiveness and medium repetition frequency (once per video request for each server). The tool chosen by the UANM platform in the case of a wired path with capacity upper bounded by 100Mbps for this measurement profile would be Pathchirp [29] as indicated in Table 2 - *predefined measurement profiles* of [1], specifically the entry MONITORING. To confirm these findings, we also performed a simulation study with the tools available as plugins in UANM. The following consideration have been drawn from this study. First, increasing the spread factor (i.e. decreasing the probing rate) does not affect the QoS linearly. The number of flows accepted increases with a larger spread factor. However, the benefits of this are nullified by the significant increase experienced in end-to-end delay. Second, as the inter-estimate time lowers so does the end-to-end delay, an inter estimate time of one second requires a high overhead of control and probe packets. Therefore, in the rest of these simulations, aimed at showing the benefits of available bandwidth estimation for server selection, we utilize Pathchirp with spreading factor equal to 1.2 and inter-estimate time equal to 5 s. We refer the interested readers to [1] and [21] for further details.

The following tests were carried out on the simulated topology. We analyze the performance of the algorithm when operating in both a steady state

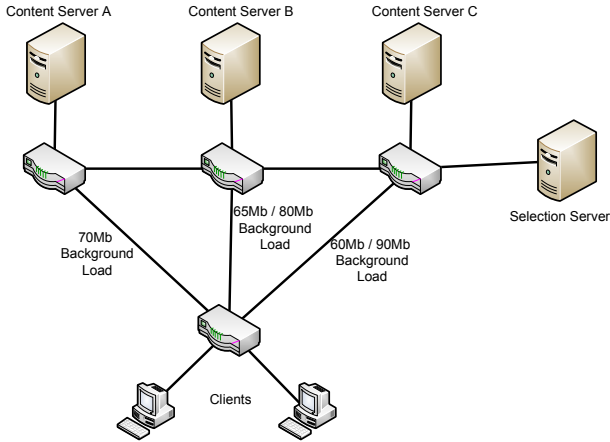


Fig. 4. Topology Used In Simulation Environment

network environment with different kinds of background traffic and also in the situation where there is a degradation in the condition of the network, such as, for example, when the background traffic increases. The number of links between the content servers and the clients access point was kept equal at 2 hops and background loads were added to the paths by using OPNET's traffic generators. These traffic generators provided a base background traffic of 70Mbps, 65Mbps, and 60Mbps on the paths of the three content servers mentioned, as shown in Figure 4. The packet sizes for the background loads were uniformly distributed with an average of 576 bytes (IPv4 MTU). To create the increase in background traffic we introduce a step change to two of the traffic generators. The 65Mbps background load is increased to 80Mbps and the 60Mbps background load is increased to 90Mbps. This is also shown in Figure 4. Overall, this reduces the available bandwidth from 105Mbps down to 60Mbps with a significant change in where the majority of that bandwidth is available.

We consider the number of requests admitted to the network, and compare the case in which available bandwidth estimates are used to analyze the connection quality, against the one where no such information is available. The results in Figure 5a depict that the use of available bandwidth estimates allows for a dynamic response to the changing bandwidth conditions within the network by lowering the number of requests admitted as the background traffic increases.

We also introduced a step function to simulate a dramatic change in background traffic to analyze how the admission control algorithm responded. The test consisted of three classes of video requests arriving at different request rates. In Figure 5b all the classes of video see a reduction in the throughput they are generating. These simulations allowed to show how, relying on the available bandwidth estimation, this approach does not need access to the intermediate topology and the video traffic is robust to external changes in the available bandwidth as it adapts the number of requests admitted.

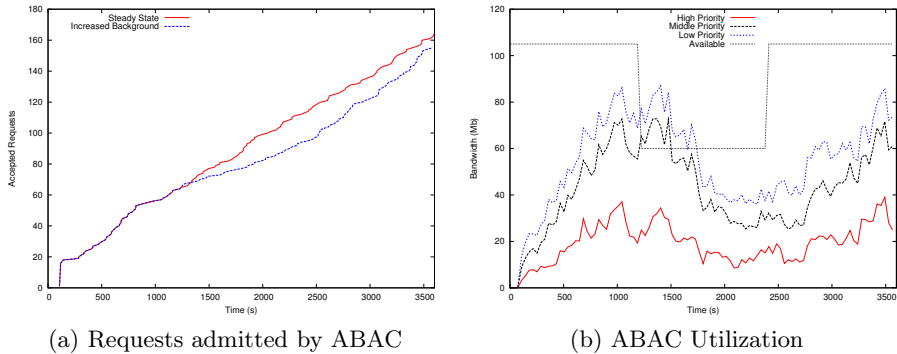


Fig. 5. ABAC Analysis

5 Conclusions

In this paper we have presented a very important metric for analyzing the quality of communication links, namely available bandwidth. We described the novel UANM platform that has been conceived for properly choose, configure, and use the large number of available bandwidth estimation tools available in literature. As available tools and techniques require tailoring to specific scenarios and require an expert operator to obtain accurate results, the platform facilitates detailed analysis of each tool under common network scenarios and can be used to identify appropriate configurations of tools to suit particular applications. Thanks to UANM, we have also shown the results of a comparison of the performance, accuracy, and intrusiveness of the mainly used tools for available bandwidth estimation, highlighting their pros and cons in different scenarios. Finally, we presented a server selection and admission control framework which bases decisions on measurements of available bandwidth between the client host and a server host. We chose to use Pathchirp with an appropriate configuration to suit our requirements, based on the analysis carried out through UANM. The framework demonstrated that using appropriately configured available bandwidth estimations can ensure that effective decisions are taken for both choosing an appropriate server to deliver video content to the client host, and also whether there is enough bandwidth available on the end-to-end path to host the video stream, without incurring any additional congestion within the network.

References

1. Aceto, G., Botta, A., Pescapé, A., D’Arienzo, M.: Unified architecture for network measurement: The case of available bandwidth. *Journal of Network and Computer Applications* (2011)
2. Aceto, G., Botta, A., Pescapé, A., D’Arienzo, M.: UANM: a platform for experimenting with available bandwidth estimation tools. In: *IEEE Symposium on Computers and Communications*, pp. 174–179 (2010)

3. Ali, A.A., Michaut, F., Lepage, F.: End-to-End Available Bandwidth Measurement Tools: A Comparative Evaluation of Performances. In: 4th Intl Workshop on Internet Performance, Simulation, Monitoring and Measurement, Austria (June 2006), <http://arxiv.org/abs/0706.4004>
4. Angrisani, L., Botta, A., Pescapé, A., Vadursi, M.: Measuring wireless links capacity. In: 2006 1st International Symposium on Wireless Pervasive Computing, p. 5 (2006), <http://dx.doi.org/10.1109/ISWPC.2006.1613635>
5. Botta, A., D'Antonio, S., Pescapé, A., Ventre, G.: BET: a hybrid bandwidth estimation tool. In: Proceedings of the 11th International Conference on Parallel and Distributed Systems, vol. 2, pp. 520–524 (2005), <http://dx.doi.org/10.1109/ICPADS.2005.103>
6. Botta, A., Pescapé, A., Ventre, G.: On the performance of bandwidth estimation tools. In: Proceedings of Systems Communications, pp. 287–292. IEEE (2005)
7. Botta, A., Dainotti, A., Pescapé, A.: A tool for the generation of realistic network workload for emerging networking scenarios. *Computer Networks* 56(15), 3531 (2012)
8. Botta, A., Pescapé, A., Ventre, G.: An approach to the identification of network elements composing heterogeneous end-to-end paths. *Computer Networks* 52(15), 2975–2987 (2008)
9. Botta, A., Pescapé, A., Ventre, G.: Quality of service statistics over heterogeneous networks: Analysis and applications. *European Journal of Operational Research* 191(3), 1075–1088 (2008)
10. Cabellos-Aparicio, A., Garcia, F., Domingo-Pascual, J.: A novel available bandwidth estimation and tracking algorithm. In: Proceedings of IEEE Network Operations and Management Symposium, pp. 87–94 (2008)
11. Cheng, X., Dale, C., Liu, J.: Understanding the characteristics of internet short video sharing: Youtube as a case study. In: ACM SIGCOMM Conference on Internet Measurement, p. 28 (2007)
12. Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., Bowman, M.: Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review* 33(3), 3–12 (2003)
13. Geni-global environment for network innovations (September 2012), <http://www.geni.net>
14. Goldoni, E., Rossi, G., Torelli, A.: Assolo, a new method for available bandwidth estimation. In: International Conference on Internet Monitoring and Protection, pp. 130–136 (2009)
15. Goldoni, E., Schivi, M.: End-to-End Available Bandwidth Estimation Tools, An Experimental Comparison. In: Ricciato, F., Mellia, M., Biersack, E. (eds.) TMA 2010. LNCS, vol. 6003, pp. 171–182. Springer, Heidelberg (2010)
16. Guerrero, C.D., Labrador, M.A.: Traceband: A fast, low overhead and accurate tool for available bandwidth estimation and monitoring. *Computer Networks* 54(6), 977–990 (2010), <http://dx.doi.org/10.1016/j.comnet.2009.09.024>
17. Hu, N., Steenkiste, P.: Evaluation and characterization of available bandwidth probing techniques. *IEEE Journal on Selected Areas in Communications* 21, 879–894 (2003)
18. IBM Corporation: An architectural blueprint for autonomic computing. *Autonomic Computing*, white paper (2006)
19. Jain, M., Dovrolis, C.: End-to-end available bandwidth: measurement methodology, dynamics, and relation with tcp throughput. *IEEE/ACM Trans. Netw.* 11(4), 537–549 (2003)

20. Johnsson, A., Melander, B., Bjorkman, M.: Diettopp: A first implementation and evaluation of a simplified bandwidth measurement method. In: Proceedings of the 2nd Swedish National Computer Networking Workshop (2004)
21. Meskill, B., Davy, A., Jennings, B.: Server selection and admission control for IP-based video on demand using available bandwidth estimation. In: IEEE Conference on Local Computer Networks (2011)
22. Meskill, B., Davy, A., Jennings, B.: Revenue-maximizing server selection and admission control for iptv content servers using available bandwidth estimates. In: Proc. IEEE/IFIP Network Operations and Management Symposium (2012)
23. Mingzhe, L., Claypool, M., Kinicki, R.: Wbest: A bandwidth estimation tool for ieee 802.11 wireless networks. In: 33rd IEEE Conference on Local Computer Networks (LCN), pp. 374–381 (October 2008)
24. Murray, M., Smallen, S., Khalili, O., Swany, M.: Comparison of end-to-end bandwidth measurement tools on the 10GigE TeraGrid backbone. In: The 6th IEEE/ACM International Workshop on Grid Computing, pp. 300–303 (2005)
25. Navratil, J., Cottrell, R.L.: Abwe: A practical approach to available bandwidth estimation. In: Proceedings of the Workshop on Passive and Active Measurement (PAM), La Jolla (2003)
26. OPNET: Discrete event smulation model library. OPNET ModelerTM (2011), <http://www.opnet.com/>
27. Paxson, V.: Strategies for sound internet measurement. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement, IMC 2004, pp. 263–271. ACM, New York (2004), <http://doi.acm.org/10.1145/1028788.1028824>
28. Prasad, R., Jain, M., Dovrolis, C.: Effects of Interrupt Coalescence on Network Measurements. In: Barakat, C., Pratt, I. (eds.) PAM 2004. LNCS, vol. 3015, pp. 247–256. Springer, Heidelberg (2004)
29. Ribeiro, V., Riedi, R., Baraniuk, R., Navratil, J., Cot, L.: pathchirp: Efficient available bandwidth estimation for network paths. In: Passive and Active Measurement Workshop (2003)
30. Seeling, P., Reisslein, M., Kulapala, B.: Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial. IEEE Communications Surveys Tutorials 6(3), 58–78 (2004)
31. Shriram, A., Murray, M., Hyun, Y., Brownlee, N., Broido, A., Fomenkov, M., Claffy, K.: Comparison of Public End-to-End Bandwidth Estimation Tools on High-Speed Links. In: Dovrolis, C. (ed.) PAM 2005. LNCS, vol. 3431, pp. 306–320. Springer, Heidelberg (2005)
32. Sommers, J., Barford, P., Willinger, W.: A proposed framework for calibration of available bandwidth estimation tools. In: Proceedings of the 11th IEEE Symposium on Computers and Communications, ISCC 2006, pp. 709–718 (2006)
33. Song, H., Zhang, Q.: Netquest: A flexible framework for large scale network measurements. IEEE/ACM Transactions on Networking 17(1), 106–119 (2007)
34. Strauss, J., Katabi, D., Kaashoek, F.: A measurement study of available bandwidth estimation tools. In: ACM SIGCOMM IMC, pp. 39–44 (2003)
35. Xu, D., Qian, D.: A bandwidth adaptive method for estimating end-to-end available bandwidth. In: 11th IEEE International Conference on Communication Systems, pp. 543–548 (2008)

Internet Topology Discovery

Benoit Donnet

Université de Liège – Liège – Belgium

Abstract. Since the nineties, the Internet has seen an impressive growth, in terms of users, intermediate systems (such as routers), autonomous systems, or applications. In parallel to this growth, the research community has been looking for obtaining and modeling the Internet topology, i.e., how the various elements of the network interconnect between themselves. An impressive amount of work has been done regarding how to collect data and how to analyse and model it.

This chapter reviews main approaches for gathering Internet topology data. We first focus on hop limited probing, i.e., traceroute-like probing. We review large-scale tracerouting projects and discuss `traceroute` limitations and how they are mitigated by new techniques or extensions. Hop limited probing can reveal an IP interface vision of the Internet. We next focus on techniques for aggregating several IP interfaces of a given router into a single identifier. This leads to a router level vision of the topology. The aggregation can be done through a process called alias resolution. We also review a technique based on IGMP probing that silently collect all multicast interfaces of a router into a single probe. We next refine the router level topology by adding subnet information. We finish this chapter by discussing the AS level topology, in particular the relationships between ASes and the induced hierarchy.

Keywords: Internet topology, traceroute, alias resolution, IGMP, MERLIN, subnet, AS.

1 Introduction

Internet is made of a vast set of heterogeneous and interconnected entities enabling the communication between millions of machines. Typically, this network is described as a graph [1] where nodes refer to IP interfaces, routers, or autonomous systems (ASes)¹ and links represent the existence of a direct connection between those nodes. This is illustrated in Fig. 1 where black dots represent router interfaces, blank shapes stand for routers, and shaded areas for ASes. The plain and dotted lines correspond to links. The router graph can be obtained when all interfaces of a router are grouped in a single identifier. This process is known as *alias resolution*. Finally, the AS level is obtained when we look only

¹ Note there are other possible levels, not shown on Fig. 1, such as the Point-of-Presence (PoP) level [2–5] or the subnet level [6–8]. This latter level will be the subject of Sec. 4, while the PoP level will not be addressed in this chapter.

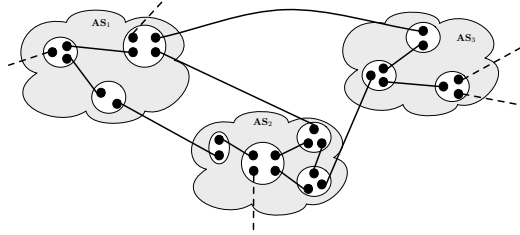


Fig. 1. The different levels of Internet topology

at ASes and the links between them (in some sense, we aggregate all routers belonging to a given AS into a single identifier, the AS number).

This infrastructure, known as *Internet topology*, makes possible the course of the information from a given node towards any other node with the help of intermediate infrastructures acting as relay for the information. This is, de facto, the backbone of a large number of applications, ubiquitous in our society, like Internet browsing, email, peer-to-peer systems, cloud computing, and many others.

Consequently, our deep understanding of the Internet topology properties (see, for instance, [1, 9–14]) and its dynamics is crucial as it impacts our capacity to maintain good performance of the network [15], to improve its efficiency and to design relevant protocols for various applications [16]. Those tasks naturally lean upon theoretical studies and simulations realized with artificial graphs obtained from models of the Internet topology [10].

However, it must be understood that the network organisation cannot be, at a given time, directly available. Its evolution is not ruled by any central authority that could have a global vision of its structure. As a consequence, the data collection can only rely on network measurements: researchers and engineers use complex procedures and tools for building Internet maps, as complete as possible, gathering so light on some Internet properties. Secondly, assuming that the data collected is correct and representative of the actual network, efforts are made for creating Internet models that are essential for simulations [10]. Lots of works have thus been done for collecting larger and larger amount of data and modeling as accurately as possible the network.

In this chapter, we investigate how Internet topology data can be collected. In particular, we focus on techniques for gathering IP interface information through *traceroute*-like probing (Sec. 2). We describe how *traceroute* [17] works, review large-scale projects using *traceroute*, and discuss *traceroute* main limitations and how they can be circumvented. We also focus on the router level of the Internet topology (Sec. 3). We describe techniques for aggregating IP interfaces of a router into a single identifier (i.e., alias resolution), those techniques being active or passive. We also discuss a recent active probing technique, *IGMP probing*, that naturally provides a router level view of the topology. We next refine the router level with subnet information (Sec. 4). Finally, we have a look at the AS level discovery (Sec. 5). In particular, we describe the ASes hierarchy and the relationships between ASes and their inference. We describe the common dataset for studying the Internet AS level topology.

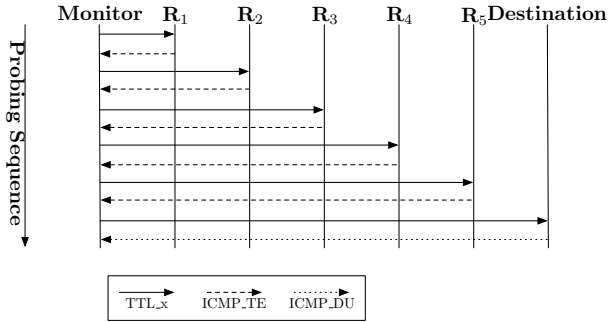


Fig. 2. traceroute example

2 IP Interface Level Discovery

The IP interface level is the lowest level of the Internet topology. It considers the IP interfaces of routers and end-hosts. All routers and some hosts have multiple interfaces and each interface appears as a separate node in this topology. The graph's links consist of the link-layer connections between nodes. These may not be point-to-point beneath IP: there may be tunnelling across lower-layer protocols, such as MPLS [18, 19], and there might be traversal of multiple layer-2 devices [20].

As the IP interface level of the Internet topology can be discovered using `traceroute`, we first review how the `traceroute` tool works (Sec. 2.1). Next, we discuss several topology mapping projects that are based on `traceroute` (Sec. 2.2). Finally, we discuss some `traceroute` limitations and how they are fixed (Sec.2.3).

2.1 traceroute

`traceroute` [17] is a networking tool for revealing the path a data packet traverses, from a machine S (the *source* or the *monitor*) to a machine D (the *destination*). `traceroute` was created by Van Jacobson in 1989. A variant of Van Jacobson's `traceroute`, the `NANOG traceroute`, is maintained by Gavron [21]. `NANOG traceroute` has additional features such as AS lookup, TOS support, microsecond timestamps, path MTU discovery, and parallel probing.

Fig. 2 illustrates how `traceroute` works. `Monitor` is the `traceroute` source, `Destination` is the destination and the R_i s are the routers along the path. The monitor sends multiple *User Datagram Protocol* (UDP) probes into the network with increasing *time-to-live* (TTL) values. Each time a packet enters a router, the router decrements the TTL. When the TTL value is one, the router determines that the packet has consumed sufficient resources in the network, drops it, and informs the source of the packet by sending back an *Internet Control Message Protocol* (ICMP) *time-exceeded* message (ICMP_TE in Fig. 2). By looking at the IP source address of the ICMP message, the monitor can learn one of the IP

addresses of the router at which the probe packet stopped. It is worth to notice that this source address is supposed to be the outgoing interface of the reply and not the interface on which the packet triggering the reply was received [22].

When, eventually, a probe reaches the destination, the destination is supposed to reply with an ICMP *destination unreachable* message (ICMP_DU in Fig. 2) with the code *port unreachable*. This works if the UDP packet specifies a high, and presumably unused, port number, i.e., above 1024.

Standard **traceroute**, as just described, is based on UDP probes. However, two variants exist. The first variant is based on ICMP. Instead of launching UDP probes, the source sends ICMP *Echo Request* messages. With ICMP **traceroute**, the destination is supposed to reply with an ICMP **echo_reply**. The second variant sends *Transport Control Protocol* (TCP) packets. The TCP **traceroute** [23] aims at bypassing most common firewall filters by sending TCP SYN packets. It assumes that firewalls will permit inbound TCP packets to specific ports, such as port 80 (HTTP), listening for incoming connections. The behavior of the **traceroute** for the intermediate routers is the same as in standard **traceroute**.

The probing method used has an impact on the collected dataset. Indeed, Luckie et al. show that there are significant differences in the topology observed following the probing method [24]. ICMP-based **traceroute** is able to reach more destinations and to collect more evidence of a greater number of AS links (after an IP-to-AS mapping [25–27]). If UDP-based probing reaches less destinations, it is however able to reveal much more IP links.

2.2 traceroute-Based Projects

traceroute is a tool easy to implement, manipulate, and deploy (standard **traceroute** is part of any operating system distribution). As such, it became the de-facto standard for probing the network, not only for collecting topology data but also as a network diagnostic tool. In this section, we review several topology mapping projects using **traceroute** for gathering data.

Archipelago [28] is CAIDA’s current measurement infrastructure (i.e., it is the successor of *skitter* [29]). *Archipelago* is based on team probing, i.e., probing monitors are grouped into teams and the measurement work is dynamically divided among team members. Currently, *Archipelago* is made of more than 50 monitors (globally distributed among commercial and research networks) grouped in three teams. The parallelization allows one to obtain **traceroute** data from all routed /24’s in about two or three days. **traceroute** performed by *Archipelago* are made using *scamper* [30], a modern **traceroute** implementation.

RIPE NCC’s *Test Traffic Measurement* (TTM) [31] measures key parameters of the connectivity between a given site and other test boxes. The TTM system performs measurements in a full mesh between roughly a hundred monitors. In addition to **traceroute** data, the TTM system also records, among others,

one-way delay², packet loss, and bandwidth. Measurements have been performed approximately once every ten minutes, starting in October 2002.

NLANR's Active Measurement Project (AMP) [32] performed active measurements connected by high performance IPv4 networks. 150 AMP monitors were deployed and take site-to-site measurements, mainly throughout the United States. Like RIPE NCC TTM, NLANR AMP avoided probing outside its own network. In addition to `traceroute`, AMP measured RTT, packet loss, and throughput. An IPv6 version of AMP performed measurements between eleven sites. AMP data collection ceased in early September 2006.

The *Distributed Internet MEasurements and Simulations* (DIMES) [33] system is a measurement infrastructure that achieves a large scale by following the model of SETI@home [34]. SETI@home provides a screensaver that users can freely install, and that downloads and analyzes radio-telescope data for signs of intelligent life. The project obtains a portion of the computing power of the users' computers, and in turn the users are rewarded by the knowledge that they are participating in a collective research effort, by attractive visualisations of the data, and by having their contributions publicly acknowledged. DIMES provides a publicly downloadable route tracing tool, with similar incentives for users. It was released as a daemon in September 2004. The DIMES agent performs internet measurements such as `traceroute` and ping at a low rate, consuming at peak 1KB/sec.

Atlas [35] is a system that facilitates the automatic capture of IPv6 network topology information from a single probing source. Atlas is based on "source-routed IPv6 `traceroute`", i.e., it performs `traceroute` on IPv6 networks and the `traceroute` can use source routing facilities. Although source routing is largely disabled in IPv4 networks, it is enabled in IPv6 networks. Source routing allows greater coverage than can ordinarily be achieved by a single `traceroute` monitor. To initiate the discovering process, Atlas relies on probing paths among a set of known addresses called *seeds*. The seeds are derived from the information in the 6Bone registry, a public database of sites and their respective address allocations. To increase probing performance without overloading the network, Atlas uses caching. For each trace, the probe engine caches the hop distance to the *via-router*, i.e., the intermediate router used for source routing. If the same *via-router* is used in a subsequent trace, then the cache distance provides the initial hop distance and alleviates the need to re-probe from the probing source to that *via-router*. Note that others tools for discovering the IPv6 network are available, such as *Dolphin* [36], *scamper* [30], *Archipelago* [28], and *SRPS* [37].

iPlane [38] is a service providing Internet path performance predictions by building an annotated map of the Internet. Measurement points are deployed on the PlanetLab testbed and the network is daily probed (`traceroute` probing). The obtained data is then postprocessed in order to provide finer grained information, such as router level topology (see Sec. 3 for details on how to build router level maps from `traceroute` data), IP-to-AS mapping, IP-to-PoP mapping, bandwidth estimation, etc.

² This is possible as each box in the system has a GPS.

Discarte [39] is an example of a new scalable probing technique that relies on existing techniques. Indeed, *Discarte* extends **traceroute**-like probing by using the *Record Route* option defined in the IP header [40]. This option is a way to record the route of an IP packet. If a router detects the Record Route option in the IP packet received and this router enables this option, the router must record in the IP header the address it uses for forwarding the packets. General usability of IP options has been investigated by Fonseca et al. [41]. They found that half of the paths drops packets with IP options but those drops occur mainly at the edge and are done by a minority of ASes.

Using the Record Route option within **traceroute** comes with several advantages, one of them being the ability to gather multiple IP interfaces of a router (and, thus, perform alias resolution) without additional probing. However, it has the drawbacks that this option is length limited (only nine IP addresses can be inserted in this IP option) and is not broadly supported by routers. In addition, *Discarte* uses a logical inference and constraint solving technique to merge Record Route and **traceroute** data. *Hynetd* [42] also uses Record Route to improve the discovering process.

Gulliver [43] is a measurement platform (currently more than 50 monitors) aiming at observing the behavior of the Internet from all over the world. Monitors are performing DNS measurements, as well as **traceroute** exploration.

2.3 Limitations

If **traceroute** is the most used tool for discovering the IP interface level of the Internet topology, it suffers from several limitations.

First, **traceroute** is *routing dependent*. This means that, when tracerouting, we are only able to observe what the Internet accepts to reveal. Thus, for instance, backup links are unlikely to be traversed by **traceroute**. The number of probing monitors and **traceroute** destinations have been subject to intensive works [44–46], in particular how the number of monitors and destinations can increase the quantity of topology information collected. If obtaining a list of **traceroute** destinations can be straightforward (it is enough to pick an IP address in each routed /24, for instance), obtaining probing monitors might not be that easy. DIMES solves this issue by proposing a “community-oriented” solution, i.e., the tool is deployed based on the community goodwill. Chen et al. suggest a measurement platform that scales with the growing Internet [47]. They embed a **traceroute** utility into a popular peer-to-peer (P2P) system and traceroutes are performed each time a P2P client is up and running. Doing this way, Chen et al. were able to probe from more than 900,000 IP addresses during the data collection period (one year and a half). Note that it can be difficult to perform reliable measurements since the P2P nodes can go down at any time without warning. Speeding up the probing process is also supposed to improve the topology vision because network dynamics could be better captured [48, 49].

Second, **traceroute** probes are subject to *load balancing*, leading to the inference of false links between IP interfaces. This drawback is explored in Sec. 2.3.1.

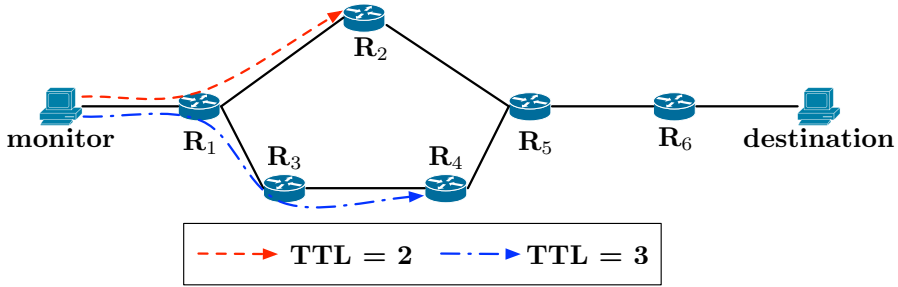


Fig. 3. Effect of load-balancing on `traceroute` exploration

Third, `traceroute` consumes a lot of network resources by repeatedly discovering the same interfaces and links. This problem is known as *redundancy* and is the focus of Sec. 2.3.2.

Fourth, it is believed that MPLS tunnels obscure topology information and `traceroute` traversing a tunnel cannot reveal the tunnel content. This problem is known as *hidden routers* as MPLS hides topology information to `traceroute`. We focus on this limitation in Sec. 2.3.3.

Fifth, `traceroute` is *unidirectional*. If we `traceroute` from A to B , it is likely the path inferred will differ from the *reverse path*, i.e., the one from B to A . We discuss this issue in Sec. 2.3.4.

Finally, not all routers respond to `traceroute` probes. Such routers are called *anonymous routers* and imply holes in the inferred paths between sources and destinations. We discuss anonymous routers in Sec. 2.3.5.

Additional `traceroute` anomalies are also described by Augustin et al. [50] and Viger et al. [51]. Finally, note that the problem of *third-party* addresses [52, 53] will be addressed in Sec. 5.5 as it only concerns AS inference from `traceroute` data.

2.3.1 Load Balancing

Load balancing for Internet paths is extensively used by ISPs for ensuring reliability of their networks and improve resource utilisation. This can be done through intra-domain routing protocols, such as OSPF [54] and IS-IS [55], supporting *equal cost multipath* (ECMP) [56], a routing strategy in which the next-hop to a given destination can occur over multiple paths. Load balancing can also be considered in the context of multihomed ISPs, for selecting which provider will receive which packet [57]. A router performing load balancing is called *load balancer*. Fig. 3 illustrates a load balancing path between a source and a destination. Router R_1 acts as load balancer as the path to “destination” can go through R_2 or through $R_3 \rightarrow R_4$.

There are three types of load balancing [58, 59]: *per packet*, *per flow*, and *per destination*. With per flow load balancing, a flow is assigned to each packet through information in the packet header, and the load balancer forwards all packets belonging to the same flow to the same interface. A flow identifier can be built, for instance, based on the classic five-tuple: source IP address,

destination IP address, source port, destination port, and protocol. Others IP field, such as ToS, can also be considered for the flow identifier. The per packet load balancing only ensures an even load on the links, making no attempt to maintain a flow. Finally, the per destination load balancing forwards all packets with the same destination to the same interface of the load balancer and can be seen as equivalent to standard routing.

The presence of load balancing implies that there are multiple routes. The historical `traceroute`, as developed by Van Jacobson (see Sec. 2.1), is sensitive to load balancing. This can lead to the inference of false links between routers, as illustrated in Fig. 3. If router R_1 is, for instance, a per packet load balancer, a `traceroute` packet with $TTL = 2$ (dashed link on Fig. 3) could reach router R_2 but, with the packet with $TTL = 3$ (dash-dotted link on Fig. 3), it could reach router R_4 , inferring so a link between R_2 and R_3 . However, such a link does not exist.

Luckie et al. evaluated the inaccuracies induced from false links inferences [60]. This evaluation has been done at two levels: macroscopic probing (i.e., `tracerouting` towards a very large set of destinations, in the fashion of Archipelago – see Sec. 2.2) and ISP probing (i.e., in the fashion of Rocketfuel [61]). Regarding macroscopic probing, the impact of false links seems to be minor while, for the latter, two third of the links were suspicious.

A new `traceroute`, called *Paris traceroute*, has been developed by Augustin et al. in order to take into account load balancing and to avoid false link inference [50]. The idea behind Paris traceroute is to control the `traceroute` packet header fields so that all probes towards a destination follow the same path. This allows one to avoid the negative effects of per flow load balancing on `traceroute` exploration. Per packet load balancing is much more difficult to mitigate due to its random nature.

Based on Paris traceroute, an algorithm, *Multipath Detection Algorithm* (MDA), for detecting and listing all routes induced by a load balancer has been proposed [62–64]. The deployment of this algorithm shows that 39% of the source-destination pairs traverse a per flow load balancer and 70% a per destination load balancer.

2.3.2 Redundancy

Donnet et al. evaluate how `traceroute` probing involves duplicate efforts [65, 66]. This is of high importance as `traceroutes` emanating from a large number of monitors and converging on selected targets can easily appear as a distributed denial-of-service (DDoS) attack. Whether or not it triggers alarms, it is not desirable for a measurement system to consume undue network resources. Duplicated effort in such systems takes two forms: measurements made by an individual monitor that replicate its own work, and measurements made by multiple monitors that replicate each other’s work. Donnet et al. term the first *intra-monitor redundancy* and the second *inter-monitor redundancy*.

On one hand, intra-monitor (shown in Fig. 4(a) with very thick arrows illustrating redundant portion of the explored graph) redundancy occurs in the context of the tree-like graph that is generated when all `traceroutes` originate

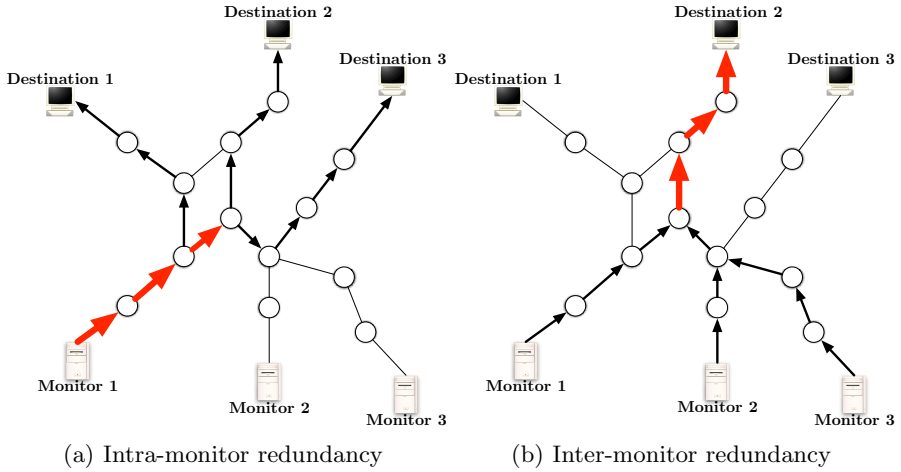


Fig. 4. traceroute redundancy [65]

at a single point. Since there are fewer interfaces closer to the monitor, those interfaces will tend to be visited more frequently. In the extreme case, if there is a single gateway router between the monitor and the rest of the Internet, a single IP address belonging to that router should show up in every one of the **traceroutes**. On the other hand, inter-monitor redundancy (shown in Fig. 4(b)) with very thick arrows illustrating redundant portion of the explored graph) occurs when multiple monitors visit the same interface.

Solutions to probing redundancy have been explored. First, *Scriptroute* [67]’s *Reverse Path Tree* (RPT) discovery tool is used to avoid the network overloading when multiple monitors probe towards a given destination. A reverse path tree is a destination-rooted tree, i.e., a tree formed by routes converging from a set of monitors on a given destination (see Fig. 4(b)). The RPT tool avoids retracing paths by embedding a list of previously observed IP addresses in the script that directs the measurements. A given monitor stops probing when it reaches a part of the tree that has already been mapped. *Scriptroute* thus can avoid inter-monitor redundancy.

Second, *Rocketfuel* [61] is a tool for mapping router-level ISP topologies. For reducing the number of measurements required, *Rocketfuel* makes use of *ingress reduction* and *egress reduction* heuristics. Ingress reduction is based on the observation that probes to a destination from multiple monitors may converge and enter a target ISP at the same node. Egress reduction is based on the observation that probes to multiple destinations may leave the target ISP at the same node.

Finally, *Doubletree* [65, 66] takes advantage of the tree-like structure of routes in the context of probing, as illustrated in Fig. 4. Routes leading out from a monitor towards multiple destinations form a tree-like structure rooted at the monitor (Fig. 4(a)). Similarly, routes converging towards a destination from multiple monitors form a tree-like structure, but rooted at the destination (Fig. 4(b)). A

monitor probes hop by hop so long as it encounters previously unknown interfaces. However, once it encounters a known interface, it stops, assuming that it has touched a tree and the rest of the path to the root is also known. Using these trees suggests two different probing schemes: *backwards* (monitor-rooted tree – decreasing TTLs) and *forwards* (destination-rooted tree – increasing TTLs).

For both backwards and forwards probing, Doubletree uses stop sets. The one for backwards probing, called the *local stop set*, consists of all interfaces already seen by that monitor. Forwards probing uses the *global stop set* of (interface, destination) pairs accumulated from all monitors. A pair enters the global stop set if a monitor receives a packet from the interface in reply to a probe sent towards the destination address.

A Doubletree monitor starts probing for a destination at some number of hops h from itself. It will probe forwards at $h + 1$, $h + 2$, etc., adding to the global stop set at each hop, until it encounters either the destination or a member of the global stop set. It will then probe backwards at $h - 1$, $h - 2$, etc., adding to both the local and global stop sets at each hop, until it either has reached the distance of one hop or it encounters a member of the local stop set. It then proceeds to probe for the next destination. When it has completed probing for all destinations, the global stop set is communicated to the next monitor. Note that in the special case where there is no response at distance h , the distance is halved, and halved again until there is a reply, and probing continues forwards and backwards from that point. Each monitor sets its own value for h in terms of the probability p that a probe sent h hops towards a randomly selected destination will actually hit that destination. Doubletree's efficiency has been largely explored [68–71].

Note that Rocketfuel's ingress and egress reduction heuristics are similar to Doubletree's forwards and backwards stopping rules. However, Rocketfuel applies its heuristics exclusively at the boundaries of ISPs, and so it does not take advantage of the redundancy reductions that might be found by paths that converge within an ISP.

2.3.3 Hidden Routers

Multiprotocol Label Switching (MPLS) [72] was designed to reduce the time required to make forwarding decisions. It is now deployed to provide additional virtual private network (VPN) services [73] and traffic engineering capability [74, 75]. To accomplish this, an IP router inserts one or more 32-bit *label stack entries* (LSE) into a packet, before the IP header, that determines the forwarding actions made by subsequent MPLS *Label Switching Routers* (LSRs) in the network. A series of LSRs connected together form a *Label Switched Path* (LSP). MPLS networks are deployed on IP routers that use a label distribution protocol [76, 77].

In an MPLS network, packets are forwarded using an exact match lookup of a 20-bit label found in the LSE. An MPLS LSE also has a time-to-live (LSE-TTL) field and a type-of-service field. At each MPLS hop, the label of the incoming packet is replaced by a corresponding outgoing label found in an MPLS switching table. The MPLS forwarding engine is lighter than the IP forwarding engine

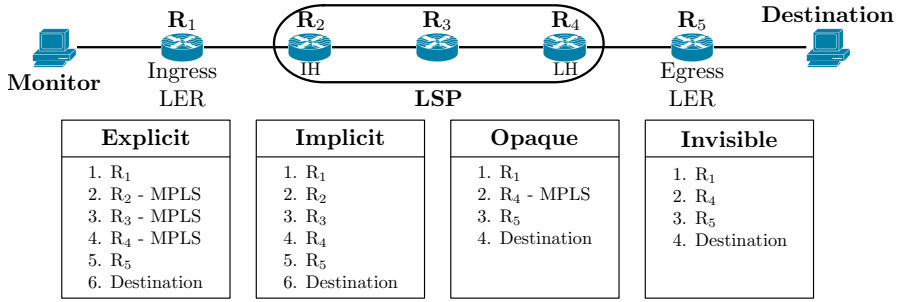


Fig. 5. Taxonomy of MPLS tunnel configurations and corresponding `traceroute` behaviors [19]

because finding an exact match for a label is simpler than finding the longest matching prefix for an IP address.

MPLS routers may send ICMP `time-exceeded` messages when the LSE-TTL expires. In order to debug networks where MPLS is deployed, routers may also implement RFC 4950 [78], an extension to ICMP that allows a router to embed an MPLS label stack in an ICMP `time-exceeded` message. The router simply quotes the MPLS label stack of the probe in the ICMP `time-exceeded` message. RFC4950 is particularly useful to operators as it allows them to verify the correctness of their MPLS tunnels and traffic engineering policy. This extension mechanism has been implemented by router manufacturers since 1999 [79], and is displayed by modified versions of `traceroute` [21] that report the label stack returned by each hop in addition to RTT values currently displayed. If the first MPLS router of an LSP (the *Ingress* Label Edge Router - LER) copies the IP-TTL value to the LSE-TTL field rather than setting the LSE-TTL to an arbitrary value such as 255, LSRs along the LSP will reveal themselves via ICMP messages even if they do not implement RFC4950. Operators configure this action using the `t1-propagate` option provided by the router manufacturer.

These two “MPLS transparency” features – RFC 4950 functionality and the `t1-propagate` option – increase the observability of otherwise opaque MPLS tunnels during IP-level topology discovery based on `traceroute`. Unfortunately, lack of universal deployment of these two features (ingress LERs that do not enable the `t1-propagate` option, and LSRs that do not support the RFC4950 ICMP extensions) means that current `traceroute`-based inference methods can cause false router-level links to be inferred and underestimates MPLS deployment in the Internet.

Based on those two MPLS transparency features, Donnet et al. [19] have proposed an MPLS taxonomy made of four classes. Fig. 5 illustrates the four classes that are

- *explicit* tunnels: both `t1-propagate` and RFC4950 are enabled. The tunnel and its internal structure are visible. Each hop within the LSP is flagged as such (as illustrated with “MPLS” in Fig. 5). Explicit tunnels have been

extensively studied by Sommers et al. [18] and are the most represented MPLS tunnels: roughly, 30% of the path traverse, at least, one MPLS explicit tunnel. This proportion of explicit tunnels has also been observed by other studies [19, 80].

- *implicit* tunnels: the router that pushes the MPLS label enables the `ttl-propagate` option but LSRs do not implement RFC4950. In this case, while the internal IP structure of the tunnel is visible, its existence as an MPLS tunnel is not revealed. Donnet et al. have proposed signature-based techniques for revealing such tunnels and demonstrated that implicit tunnels are three times less prevalent than explicit ones.
- *opaque* tunnels: LSRs implement RFC4950 but the ingress LER does not enable the `ttl-propagate` option. Only the router that pops the MPLS label reveals a LSE and the internal structure of the LSP is hidden. In Fig. 5 the opaque tunnel hides two LSRs (R_2 and R_3), allowing an erroneous link to be inferred between R_1 and R_4 . Donnet et al. have also proposed a technique for inferring the length of opaque tunnels (i.e., the number of hidden routers in the LSP). They also suggested that opaque tunnels are very infrequent.
- *invisible* tunnels: the ingress LER does not enable the `ttl-propagate` option and RFC4950 is not implemented by the router popping the MPLS label. In Fig. 5, two IP hops are hidden and the last router of the MPLS path does not flag itself as part of an LSP. Again, a link between R_1 and R_4 is erroneously inferred. Up to now, there is no technique for revealing and quantifying invisible tunnels.

Based on observation made by Sommers et al. [18] and Donnet et al. [19], MPLS is a frequent feature in the Internet but is not a brake to Internet topology discovery (thanks to RFC 4950 and `ttl-propagate` option). Only opaque and invisible tunnels are a problem but it seems they are infrequent. If for the moment invisible tunnels cannot be revealed and quantified, the actual impact of opaque tunnels on topology discovery must still be evaluated.

2.3.4 Unidirectionality

`traceroute`, as defined in Sec. 2.1, is unidirectional. It means that it is only able to capture the path from the `traceroute` source towards a given destination but without providing any information on the path from the destination to the `traceroute` source itself (i.e., the *reverse path*). Said differently, `traceroute` gives the path from the source to anywhere, but not the path from anywhere to the `traceroute` source. As routing is asymmetric [82], both paths (i.e., one-way and reverse) are likely to be different. This situation is illustrated in Fig. 6 in which the `traceroute` from the source to a given web server gives a path traversing AS₉, AS₁, AS₄, and AS₇. On the contrary, the path from the web server to the source traverses AS₇, AS₆, AS₈, AS₁₂, and AS₉.

Reverse traceroute [81] has been proposed to overcome this fundamental drawback. Reverse traceroute builds the reverse path incrementally, using a set of controlled vantage points and several measurement techniques. First, vantage points are used to measure the path from themselves towards the `traceroute`

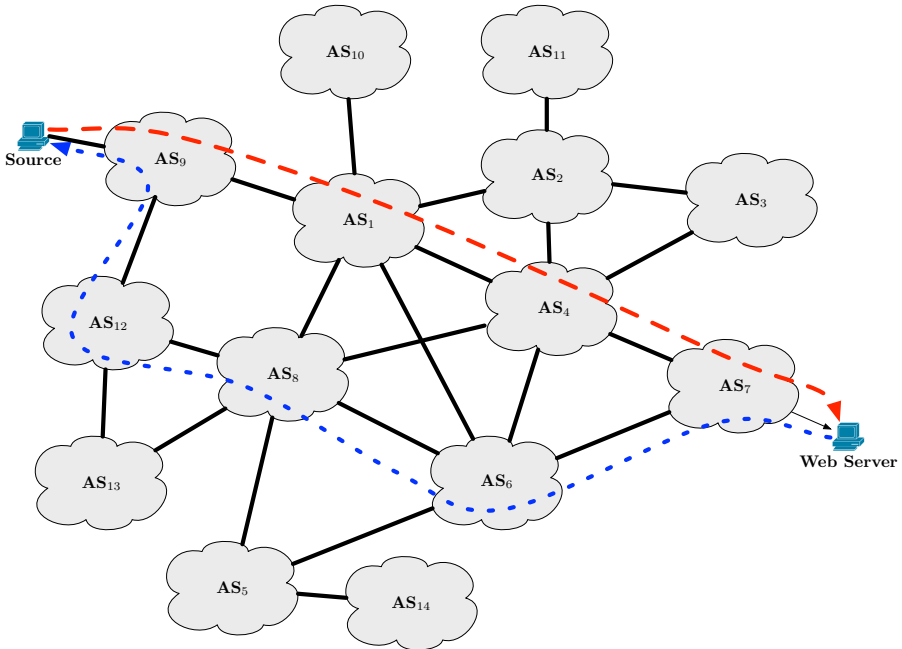


Fig. 6. traceroute unidirectionality and reverse path

source. Tree-like structure of routes is considered for avoiding probing redundancy. The set of paths collected is used as bootstrap for incrementally building the reverse path, starting from the `traceroute` destination and hop-by-hop back to the `traceroute` source. This incremental process is stopped when encountering a router belonging to the set of routes collected at first. This process can be seen as somewhat equivalent to Doubletree’s local stop set (see Sec. 2.3.2).

Reverse traceroute considers three measurement mechanisms for building the reverse path. First, globally, Internet routing is destination-based. This means that Reverse traceroute should be able to capture the reverse path one hop at a time and, next, glue all hops together. Second, several IP options, such as Record Route [40] and IP timestamp [83], are used to identify the various hops along the path. Finally, *IP spoofing* is used to overcome limitations in IP timestamp support. IP spoofing, first described by Morris [84] and deeply discussed by Bellovin [85], is one of the common tools used by hackers to perform attacks. It allows the attacker to hide his identity by forging the source IP address of packets. Instead of carrying the source IP address of the machine the packet comes from, it contains an arbitrary IP address that is selected either randomly or intentionally. Despite various techniques for avoiding IP spoofing (see, for instance, ingress filtering [86, 87], hop count [88], probabilistic marking [89], or hash-based traceback mechanisms [90]), a large-scale study has shown that IP spoofing is still widely possible [91]. Based on measurements distributed throughout the

world, Beverly and Bauer find that approximately one-quarter of the observed addresses, netblocks, and autonomous systems (AS) still permit full or partial spoofing [91].

2.3.5 Anonymous Routers

Unfortunately, the `traceroute` behavior explained in Sec. 2.1 is the ideal case. A router along the path might not reply to probes. In order to avoid waiting an infinite time for the ICMP reply, the `traceroute` monitor activates a timer when it launches the probe. If the timer expires and no reply was received, then, for that TTL, the machine is considered as *non-responding* and the router is flagged as “*” in the `traceroute` output. Such a router is also called an *anonymous router*. The task of identifying all “*” belonging to the same router is known as *anonymous router resolution*. It has been shown that this process is NP-complete [92].

Gunes and Sarac identify five reasons for a router for being anonymous [93]:

1. The router is configured to ignore all `traceroute` requests, i.e., it never sends back an ICMP *tll exceeded* packet.
2. The router applies ICMP rate limiting and is anonymous if the incoming `traceroute` queries rate is above the preset threshold.
3. The router is configured to ignore `traceroute` queries when it is congested. Otherwise, it responds to queries.
4. A border router might be configured to filter all outgoing ICMP packets coming from its domain. As a consequence, all routers belonging to this domain are anonymous.
5. The router that replies with a non-publicly routable IP address [94] should be considered as anonymous as a given non-publicly routable IP address can be used by several routers (i.e., there is no uniqueness guarantee).

It is worth to notice that anonymous router resolution cannot be done, generally, during the probing time. This process is, typically, done after the data has been collected. We can thus see anonymous router resolution as a passive process. Historically, simple solutions to anonymous routers have been proposed. For instance, Cheswick et al. [95] stop tracerouting towards a destination once an anonymous router is encountered. This approach has the drawback of potentially missing useful information. Broido and claffy [96] replace anonymous routers with arcs (e.g., the route portion $R_i \rightarrow * \rightarrow R_j$ is replaced by $R_i \rightarrow R_j$) or with a unique identifier in order to consider each anonymous router as a unique node in the topology. Such a solution can lead to inaccuracies in the resulting topology. Finally, Bilir et al. [97] compress successive anonymous routers between two nodes into a single identifier. This solution has a limited scope.

More recent solutions to anonymous router resolution are based on optimization problem [92], on heuristics on link delays or neighbor matching [93], and on graph data mining [98].

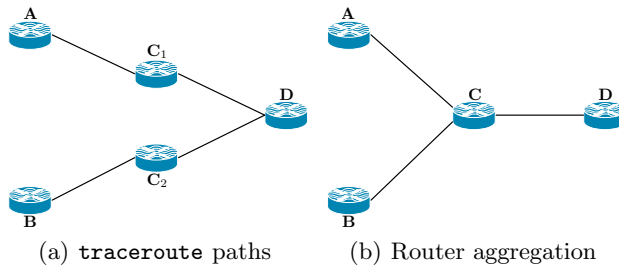


Fig. 7. Alias resolution principle

3 Router Level Discovery

The router level is the second level of the Internet topology. Nodes in the graph are routers themselves, meaning that all IP interfaces of a router collected with `traceroute` have been aggregated into a single identifier. This aggregation, historically, is made using *alias resolution* (Sec. 3.1). This alias resolution is either made at the same time of probing IP interfaces, requiring additional probing to `traceroute` (active techniques), or after the data collection by analyzing the resulting graph (passive techniques).

Recently, a new technique has been developed and is based on *IGMP probing* (Sec. 3.2). This technique has the advantage of collecting, in a single probe, all multicast interfaces of a router. This means that alias resolution is not anymore required.

3.1 Alias Resolution

The router level topology can be seen as an aggregation of the IP interface level, i.e., the summary of all the IP addresses of a router into a single identifier. The summary technique is called *alias resolution* and is illustrated in Fig. 7. As explained in Sec. 2.1, `traceroute` lists interfaces addresses of a path and identifies, in our example, interfaces A, B, C₁, C₂ and D (Fig. 7(a)). Alias resolution clusters all interfaces of a router into a single value to reveal the true topology. As shown in Fig. 7(b), interfaces C₁ and C₂ are aliases. Based on synthetic topologies, Gunes and Sarac show that the accuracy of alias resolution has an important effect on the observed topological characteristics such as, for instance, the number of nodes and edges or the average node degree [99]. In this section, we describe the currently existing approaches for alias resolution.

3.1.1 Active Methods

Active methods for alias resolution are based on several techniques: address based method, IP identifier based method, DNS, Record Route IP option, and timestamps. We describe these techniques in this section.

The *address based method* is described in RFC 1122 [100]. The principle is simple: the source sends a UDP probe with a high port number to the router

interface X . If the source address of the resulting “Port Unreachable” ICMP message is Y , then X and Y are aliases for the same router. The drawback of this solution is that some routers do not generate ICMP messages, making alias resolution impossible. This technique has been implemented in many tools, such as *iffinder* [101] and Mercator [2].

The *IP identifier based method* has been implemented in *Ally*, *Rocketfuel*’s alias resolution component [61]. *Ally* is based on the ID field of the IP header, a 16-bit field used to identify the fragments of one datagram from those of another [40]. The ID value is supposed to be unique for a given (source, destination) pair and protocol during the time the packet could be alive in the network. The basic idea of the IP identifier based method is the following: send a UDP probe packet with a high port number to the two potential aliases. The “Port Unreachable” ICMP responses are encapsulated within IP packets and, so, each one includes an *IP identifier* (x and y). Then, one sends a third packet to the address that responded first. Assume that z is the IP identifier of the third response and x was the IP identifier of the first response. If $x < y < z$ and $z - x$ is small, the addresses are likely aliases. This method, like the address based method, works only if a router responds to probes. Further, it is network resource greedy as it requires $O(n^2)$ probes to infer aliases among n IP addresses.

To overcome this network resource limitation, Bender et al. introduce *Radar-Gun* [102]. Basically, *RadarGun* models the IP ID as a counter over time and infers the rate at which the counter increases. This rate is named the *velocity* of a router and is typically close to a straight line. The distance between pairs of lines is computed and the pairs with a small distance are labeled as aliases.

MIDAR (Monotonic ID-Based Alias Resolution) [103] is designed for performing alias resolution, using the IP identifier technique, for a very large-scale of targets (on the order of several millions). The comparison of IP identifiers is, here, based on monotonicity instead of proximity. The scalability is achieved by considering multiple vantage points, multiple probing methods, and a sliding-window probe scheduling.

The *DNS based method* considers similarities in router host names and works when an AS uses a systematic naming scheme for assigning IP addresses to router interfaces. This method is especially interesting as it can work even if a router does not respond to probes directed to itself. *Ally* uses this technique against unresponsive routers with the help of the *Rocketfuel*’s name DNS decoder. *AROMA* [104] also combines DNS based method and *Ally*’s technique. However, it has been shown that DNS can introduce errors due to misnaming, leading so to bad alias resolution [105].

The *TTL-limited with record route option method* has been proposed by Sherwood and Spring [39]. The idea is to perform a standard `traceroute` with the *Record Route* (RR) IP option enabled [40]. Note that the addresses discovered by `traceroute` and RR do not overlap as RR records the outgoing interface while the *time exceeded* message solicited by `traceroute` comes from the ingoing interface. This technique works only in networks where routers support the

RR option, which is not necessarily the case in modern networks. This solution is computationally expensive [106].

Palm Tree [107] takes a list of IP addresses from a target network and aims at identifying IP aliases among IP address of this target network. The alias resolution technique applied by Palm Tree is based on common practice for assigning IP addresses [108]. Palm Tree must be seen as a complementary tool to existing techniques.

The last technique, proposed by Sherry et al. [109] and implemented in *motu* [110], is based on the *IP timestamp option* [83]. Originally, the timestamp option has been introduced for measuring the one-way delay of Internet links. A router receiving an IP packet with the IP timestamp option is supposed to provide, in the option, a timestamp consisting of milliseconds since midnight UTC. There are several sub-types of IP timestamp option: (i), “timestamp only” for which the router writes the timestamp in the option; (ii), “timestamp with ID” for which the router writes the timestamp in the option preceded with its IP address; (iii), “prespecified timestamp”, in which IP addresses of routes are prespecified in the IP packet. A router inserts its timestamp in the option only if its IP address has been prespecified by the packet sender.

For testing if A and B are aliases, Sherry et al. send multiple ICMP *Echo Request* probes with the prespecified timestamp option enabled. If A and B are aliases, the router should record timestamps for both A and B with consistent values. If timestamps in the replies are consistent, further investigations are made to ensure both IP addresses are aliases.

The large-scale applicability of the IP timestamp option has been evaluated and it has been shown that using this option is only effective for 12.9% of the destinations [111]. Note that it seems that using the IP timestamp option is a little bit more effective in the context of reverse traceroute (see Sec. 2.3.4).

Routers behaviors regarding active alias resolution have been recently analyzed [112]. Direct probing based on ICMP packets with IP identification provides the best identification ratio.

3.1.2 Passive Methods

On the contrary to active methods, passive methods do not require additional probing. The alias resolution is done offline, after the `traceroute` data has been collected. Those passive techniques are based either on graph methods, either on IP addresses and subnets assignments.

The *graph based method* extracts from `traceroute` outputs a graph of linked IP addresses in order to infer likely and unlikely aliases [113]. It is based on two assumptions: (1) if two IP addresses precede a common successor IP address, then they are likely to be alias, and (2) two addresses found in a same `traceroute` are unlikely to be aliases. This method is mainly used as a preprocessing step to reduce the number of probe pairs for an active probing approach, such as the address and IP identification based methods.

The second method is the *Analytical Alias Resolver* (AAR) introduced by Gunes and Sarac [114]. Given a set of path traces, AAR utilizes the common IP address assignment scheme to infer IP aliases within the collected path traces.

However, AAR assumes point-to-point links to resolve IP aliases on a given pair of path traces between two vantage points and completely ignores multi-access links.

The *Analytic and Probe-Based Alias Resolver* (APAR) [115] is an extension of AAR to overcome its limitations. It uses a set of inference rules, based on identifying the subnets linking routers and then aligning `traceroute` paths using those subnets. *kapar* [106] is an optimized implementation of APAR that overcomes APAR’s scalability issues.

3.2 IGMP Probing

Historical alias resolution techniques, discussed in Sec. 3.1, come with inherent limitations. Indeed, as they are based on `traceroute` data, they naturally inherit from `traceroute` limitations (see Sec. 2.3) and biases. Further, additional probing or analysis is also biased. This can lead to a high proportion of false aliases, giving so an inaccurate vision of the router level topology.

Recently, IGMP probing has been considered for collecting router level data [20]. Although it is limited to multicast-enabled routers and unfiltered networks (Sec. 3.2.4), IGMP probing comes with the advantage of silently collecting all multicast interfaces of a router in a single probe. IGMP probing is made possible with `mrinfo` (Sec. 3.2.1), `mrinfo-rec` (Sec. 3.2.2), and MERLIN (Sec. 3.2.3).

3.2.1 `mrinfo`

`mrinfo` [116] messages use the Internet Group Management Protocol (IGMP [117]). IGMP was initially designed to allow hosts to report their active multicast groups to a multicast router on their local area network (LAN). Most IGMP messages are sent with a `time_to_live` of 1. However, the Distance Vector Multicast Routing Protocol, DVMRP, has defined two special types of IGMP messages that can be used to monitor routers [118]. Although current IPv4 multicast routers do not use DVMRP anymore, they still support these special DVMRP messages. Upon reception of an IGMP `ASK_NEIGHBORS` message, an IPv4 multicast router replies by sending an IGMP `NEIGHBORS_REPLY` message that lists all its multicast enabled local interfaces³ with some information about their state. Cisco and Juniper routers also report in the IGMP `NEIGHBORS_REPLY` message the version of their operating system. Fig. 8 shows an example of the usage of `mrinfo` to query the router `R2`, `1.1.0.2` being the responding interface of `R2`. `mrinfo` reports that this router is directly connected to `R0` (through interface `1.1.0.1`). We can also notice that `R2` is connected to routers `R5` and `R6` through an L2 network (labeled “switch” in Fig. 8) because interface `1.1.2.3` appears twice in the `mrinfo` reply (see bold text in Fig. 8). Finally, `mrinfo` reports that interface `1.1.3.1` has no multicast neighbor because the right IP address is equal to `0.0.0.0` (or is directly connected to a LAN,

³ It has been reported that a router may reply to a `ASK_NEIGHBORS` message through one of its purely unicast interface [119]. In such a case, the set of collected interfaces is not only multicast.

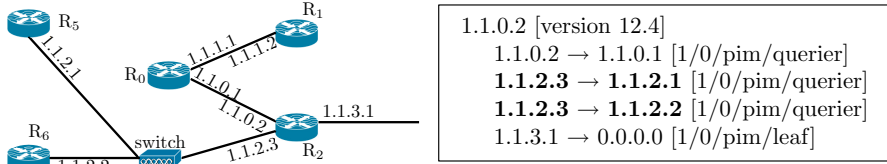


Fig. 8. `mrinfo` example [20]

as indicated by the “leaf” keyword). All this information is obtained by sending a single IGMP message. In practice, `mrinfo` provides information similar to the output of a `show` command on the router’s command line interface.

3.2.2 `mrinfo-rec`

Mérindol et al.’s approach in probing the network with `mrinfo` is recursive and is called `mrinfo-rec` [20]. Initially, `mrinfo-rec` is fed with a single IP address corresponding to the first router attached to the `mrinfo-rec` vantage point. `mrinfo-rec` probes this router and recursively applies its probing mechanism on all the collected IP addresses. These recursive queries stop at unresponsive routers or when all discovered routers have been queried. The same process is run every day. It is worth to notice that an address not replying to an `mrinfo` probe during a given day will not be queried the days after except if it appears again in a list of captured addresses.

To illustrate this behavior, let us apply it on the topology depicted in Fig. 8. `mrinfo-rec` receives, as input, an IP address belonging to router R_0 . From R_0 , `mrinfo-rec` collects a set of neighbor IP addresses, i.e., $\{1.1.1.2, 1.1.0.2\}$. For all IP addresses in this set that were not previously probed, `mrinfo-rec` sends an IGMP `ASK_NEIGHBORS` message and, if the probed router replied, it again runs through the set of neighbor IP addresses collected.

3.2.3 MERLIN

Initial implementations of `mrinfo` and `mrinfo-rec` suffer from several issues and limitations [119]. First, there is a lack of support for IGMP-fragmented `NEIGHBORS_REPLY` messages as `mrinfo` is unable to deal with multiple received packets. It only processes the first packets (there is no continuation flag forcing the wait for the remaining fragments). Mérindol et al. have observed this behavior on large degree CISCO routers. Second, `mrinfo` is unable to multiplex IGMP-based measurements. The initial version of `mrinfo` sends its IGMP query, then waits for a possible reply during a given time. Possibly it performs several retries if no response has been collected within the previous time frame. Further, IGMP does not consider port and query numbers to multiplex incoming/outgoing connections. This leads to scalability issues when performing large-scale IGMP probing.

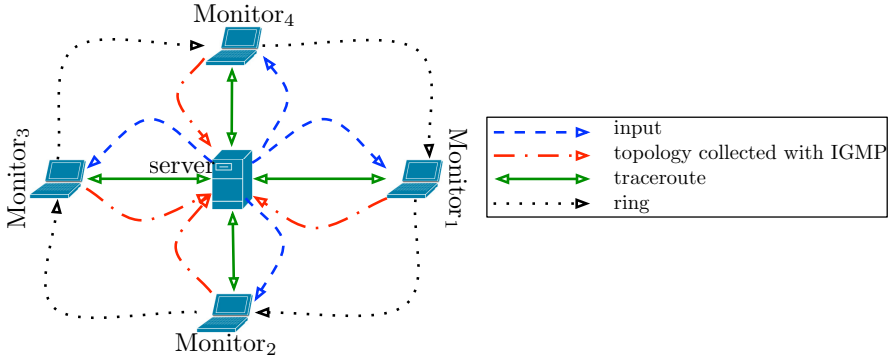


Fig. 9. MERLIN global overview [120]

MERLIN [119] has been introduced to overcome these limitations. MERLIN decouples the sending and receiving processes in order to avoid the use of timers between queries and replies and improve the probing efficiency. Furthermore, all replies having the same source IP address are considered as part of a largest message in order to re-assemble IGMP fragmented packets of a given router. If `mrinfo` and `mrinfo-rec` were probing the entire Internet, MERLIN has been designed to focus on ASes.

If MERLIN is still based on recursive IGMP probing, it also improves `mrinfo` and `mrinfo-rec` by probing from several vantage points, each one being managed by a central server [120], in order to increase the exploration coverage while limiting the probing redundancy. Fig. 9 illustrates how MERLIN vantage points are organized around the central server. The “input” is the initial set of destinations provided to each vantage points. The MERLIN vantage points are organized in a ring, a vantage point probing after its predecessor in the ring. In addition, MERLIN makes use of `traceroute` to discover active addresses (typically in targeted ASes) in order to circumvent the recursion limitations. It has been shown that this multiple vantage points probing increases the amount of information collected by the IGMP probing [119].

3.2.4 IGMP Probing Limitations

The first limitation of IGMP probing is inherent to the technique itself: only multicast enabled network can be probed. This limits thus the scale of the topology that can be collected.

When probing a multicast enabled AS with IGMP probing, one expects obtaining its complete *backbone* as it should be entirely multicast to ensure the correct multicast tree establishment by the PIM multicast routing protocol [121]. By multicast backbone, one means the AS areas where links and routers providing connectivity to non-multicast customers or peers are pruned. Unfortunately, some routers do not reply to IGMP probes sent by MERLIN, leading

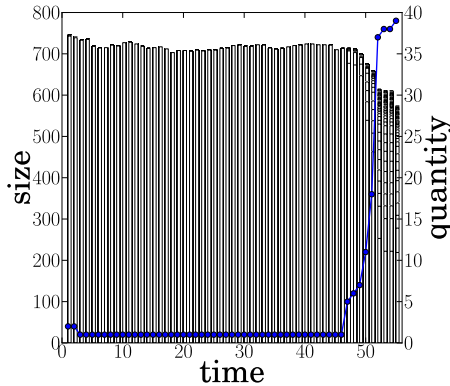


Fig. 10. Connected components evolution over time (AS1239) [120]

to an anonymous behavior that is similar to the one observed with `traceroute` (see Sec. 2.3.5). This phenomenon is called *IGMP filtering*. As a consequence, the topology obtained using solely IGMP probing is incomplete and disconnected: the collected IGMP graph exhibits a set of disjoint components. Fig. 10 illustrates this problem on the Sprint Tier-1 AS.⁴ The horizontal axis shows IGMP probing snapshots over 56 weeks (one snapshot per week) between May 2004 and December 2008. The left vertical axis provides the size of each connected component and must be read in conjunction with stacked bars. The right vertical axis gives the quantity of connected component plotted in a stacked bar and must be read with the line. While at the beginning, one was able to capture a single large connected component (more than 700 connected nodes), the number of connected components starts exploding in 2008: up to 38 connected components, a lot of them being made of only 2 nodes. The explanation of this degradation is the progressive introduction of IGMP filtering in the network. Indeed, the number of connected components increases with the number of non-responding routers.

IGMP filtering is of two kinds [122]: some multicast routers do not reply to IGMP probes (*local filtering*) while some others do not forward IGMP messages (*transit filtering*). While the second problem can be reduced with the use of multiple vantage points in a cooperative distributed platform [120], the first one is more challenging as it impacts the collected topologies. Indeed, multicast routers that do not respond to IGMP probes may divide the resulting collected multicast graph into disjoint components.

MERLIN tries to limit the effect of local filtering by applying `traceroute` and alias resolution for glueing together disconnected components [122]. This reconnection procedure is achieved by the central server, once IGMP components have been collected.

⁴ The same phenomenon can be observed in others ASes (Tier-1, Transit, and Stub).

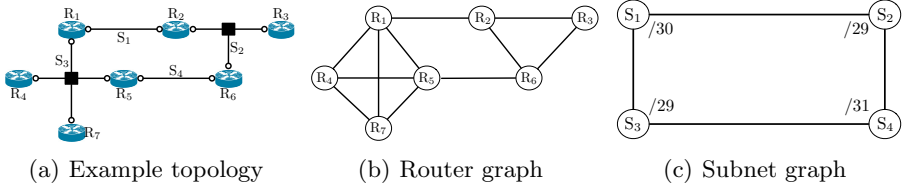


Fig. 11. How a network can be represented as a router graph and a subnet graph [124]

4 Subnet Level Discovery

A *subnetwork* (or, simply, *subnet*) refers to a set of devices that are on the same connection medium and that can communicate directly with each other at the link layer [123].

The subnet level is a way to enrich router level maps with subnet level connection information [6]. Subnet discovery presents some similarities with alias resolution (see Sec. 3.1). Indeed, alias resolution follows the goal of aggregating IP addresses (appearing in various traces) of a router into a single identifier. Similarly, subnet detection aims at identifying multiple links (appearing to be separate) and at combining them to represent their single hop connection medium (point-to-point or multi-access) [8].

Fig. 11 illustrates the concept of subnet. Fig. 11(a) provides the groundtruth topology for our example. This topology is made of seven routers, some of them being connected through layer-2 devices (the black square – see for instance the connection between R_4 and R_5). Fig. 11(b) gives the router level graph view of the topology. Finally, Fig. 11(c) aggregates all routers (and layer-2 devices) belonging to the same subnetwork into a single identifier and connects a subnet to others if they communicate with each other.

4.1 Inference Techniques

In the fashion of alias resolution (see Sec. 3.1), subnet inference can be divided into two kinds of methods: passive and active techniques.

4.1.1 Active Methods

traceNET [6] attempts to collect a subnet at each router on the same path. *traceNET* works iteratively and starts by creating a temporary /31 subnet from a given interface. It then grows the subnet by decreasing prefix lengths (i.e., /30, /29, etc.). For each subnet, *traceNET* probes the potential IP addresses within the subnet range to ensure that those IP addresses are assigned to IP interfaces. Note that this decision is taken based on heuristics. If one of the heuristics is not met, the IP address is declared as not belonging to the subnet under exploration. The growing process is then stopped and the subnet is stepped back to its last valid state.

In some sens, `traceNET` works as traceroute: it is designed to detect subnets on a given path between a source and a destination. `traceNET` is thus subject to some traceroute limitations, such as routing. In addition, in order to be efficient, `traceNET` requires several vantage points. `exploreNET` [124], a `traceNET` extension, has been introduced to mitigate those drawbacks. In particular, `exploreNET` is able to discover individual subnets rather than subnets on an end-to-end path. `exploreNET` also presents techniques for sampling subnets in the target domain and their global characteristics (such as mean subnet degree, subnet prefix length distribution, etc.) [7].

4.1.2 Passive Methods

Although it is not its primary goal, IGMP probing (see Sec. 3.2) allows one to detect subnets [14]. The subnet inference requires to post-process the collected data by following three rules:

- *Symmetry rule.* All routers attached to the potential subnet should have the same view. On Fig. 8, router R_2 is connected to the same subnet as R_5 and R_6 through a layer-2 device. When probing R_5 and R_6 with `mrinfo`, R_2 must also appear in their `mrinfo` output.
- *Querier rule.* In a normal case, only one router per layer-2 network must be tagged as the IGMP “querier” (i.e., it won the querier election on the subnet [125]: it has the greatest IP address on the subnet). For instance, on Fig. 8, as interface `1.1.2.3` is tagged as “querier”, interfaces `1.1.2.1` of R_5 and `1.1.2.2` of R_6 should not be tagged as such.
- *Subnet mask rule.* The validity of the minimum mask covering all IP addresses in the subnet is verified.

In addition, IGMP probing can provide information on the technology used in the subnet, such as ATM cloud, etc. However, the limit of IGMP probing for revealing subnet is that `mrinfo` is only able to detect subnetworks involving at least three routers.

Gunes and Sarac suggest that subnet inference can be done once data has been collected (using `traceroute`) and alias resolution has been done [8]. IP address assignment practices [126, 127] induce subnet relationships or formations. Candidate subnets are thus formed, from the data, by grouping into a subnet address range address prefix of length $/x$. Smaller subnets ($/x$, $(x+1)$, \dots , $/31$) are next recursively created. In the fashion of IGMP probing, a set of rules is defined for verifying candidate subnets:

- *Accuracy rule.* IP addresses in a subnet should appear next to each other each time they appear in the same trace.
- *Distance rule.* IP addresses from a given subnet should be at similar distances to a vantage point.
- *Completeness rule.* Candidate subnets having less than a quarter of their IP addresses present in the data set should be ignored.
- *MaxFit rule.* Candidate subnets that are a subset of a larger one must be ignored after assessing the previous rules.

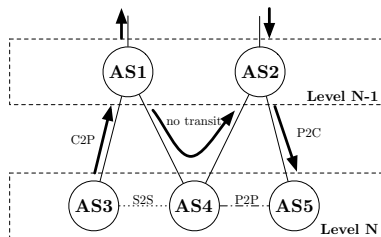


Fig. 12. AS relationships

5 AS Level Discovery

An *Autonomous System* (AS) is either a single network or a group of networks that is under the control of a single administrative entity, typically an ISP or a very large organization (for instance, a university, a business enterprise or division) with independent connections to multiple networks. An AS is also sometimes referred to as a *routing domain*. Each AS is identified by a unique 16-bit number assigned by the Internet assigned numbers authority (IANA).⁵

In this section, we describe the AS relationships (Sec. 5.1) before discussing the induced AS structure (Sec. 5.2). We next describe the various data sources for inferring AS level topology (Sec. 5.3). We describe techniques for inferring AS relationships (Sec. 5.4) and, finally, discuss their limit (Sec. 5.5).

5.1 AS Relationships

In the Internet AS topology graph, an edge between two ASes (nodes) represents a business relationship which results in the exchange of Internet traffic between them. An AS can have one or more relationships with different kinds of neighboring ASes. Each relationship may correspond to several distinct physical links [20].

On one side, an AS' *access links* connect to customer networks. Customer networks buy Internet connectivity from the AS. On the other side, *peering links* connect to transit providers from which it buys its own connectivity. Peering links also connect to private peers with which exchange of traffic is negotiated without exchanging money as a way to avoid sending traffic through a provider. No transit traffic is allowed through peering links; only traffic with the peer or its customers is permitted. These are the most observed relationships in the network and are usually referred to as the *provider-to-customer* (p2c), *customer-to-provider* (c2p) and *peer-to-peer* (p2p) relationships. A recent analysis has shown that p2p relationships between adjacent Tier-1 ASes are redundant, i.e., the connections between those ASes involve several physical links [20].

⁵ It is worth to notice that, since December 1st, 2006, the AS Number Registry has been expanded to 32 bit-number space [128].

A less common relationship found in the Internet is called the *sibling-to-sibling* (s2s) relationship. This relationship generally resides between two ASes of a same company. The key difference with peering is that siblings exchange all kinds of traffic, not only between their respective customers. An s2s relationship covers everything except the p2c, c2p and p2p relationships. It appears in various cases such as when two ASes act as backups for each other, or when two ISPs merge and decide to become siblings instead of merging into a single AS which can be very complex. Two peering ISPs have a special agreement for specific prefixes for which they transit all kinds of traffic for each other. Fig. 12 illustrates those AS relationships.

These relationships have a major impact on routing in the Internet, as shown by Tangmunarunkit et al. [129]. Inside an AS, routing uses *hop-count* as a metric, but because intra-domain protocols support hierarchies, the resulting paths are not always the shortest in terms of *hop-distance*. Between ASes, routing is determined by policy. Many Internet path lengths thus may also benefit from a detour [130, 131] which would incur more router-level hops than shortest-router-hop path routing. For simulation purpose, it is therefore most appropriate to model the network with policy-based routing rather than AS shortest path-based routing.

5.2 AS Hierarchy

Relationships discussed in Sec. 5.1 suggest the existence of an AS hierarchy [133]. This hierarchy is described in Fig. 13.

Following Subramanian et al. [133] nomenclature, we can distinguish three kinds of AS. First, the *Tier-1* ASes do not have upstream provider of their own. Typically, a Tier-1 has a national or international backbone and there are full p2p connections between them. On Fig. 13, Tier-1 ASes are on the top of the hierarchy and labeled as “National Backbone Operators”. There are a few Tier-1 ASes (roughly 12-20), such as UUNET, Sprint, or Level3. Second, the *Tier-2* ASes (or, simply, the *Transit* ASes) provide transit services to downstream providers. A transit AS needs, at least, one provider of its own. The Internet counts a few thousand Transit ASes. Those Transit ASes are located in the middle of the hierarchy in Fig. 13 and labeled as “Regional Access Providers” and “Local Access Providers”. Finally, the *Stub* ASes do not provide transit services to others. They are connected to one or more upstream providers. Stub ASes constitute the vast majority of ASes (i.e., 85-90%). They are located at the bottom of the hierarchy in Fig. 13 and labeled as “Customer IP Networks”.

5.3 Data Sources

Two sources of AS level topology data are available: *Internet registries* and *BGP routing information*. This section describe these two sources [134] along with their advantages and limitations.

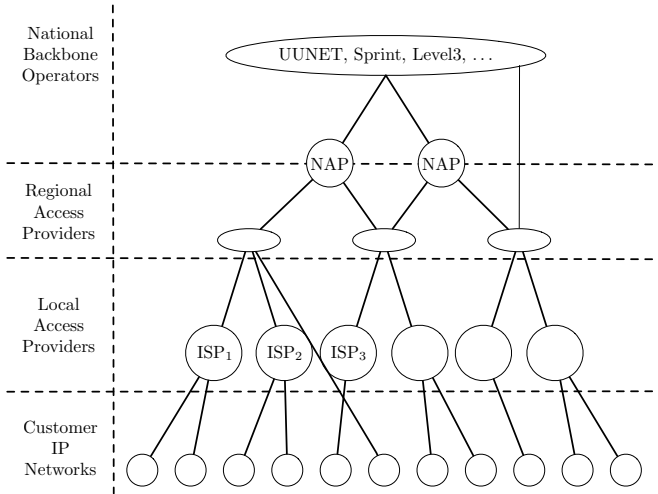


Fig. 13. Traditional AS hierarchy [132]

5.3.1 Routing Registry Information

Many publicly-available registries share information about the Internet and its topology. *Regional Internet Registries* [135] are organizations responsible for allocating AS numbers and IP address blocks, all of which are accessible using the WHOIS protocol [136]. *Internet Routing Registry* (IRR) [137] is another group of databases maintained by several organizations and containing documented routing policies. These policies are available through the WHOIS protocol and are expressed in the *Routing Policy Specification Language* (RPSL) [138].

Topology discovery using Internet registry information has several advantages. Firstly, the access is simpler and more efficient to implement than active method probing, such as those described in Sec. 2. Indeed, they do not have to explore the network to obtain the topology and the information is grouped at specific locations. Secondly, they provide high-level information such as routing policies which are otherwise more difficult to obtain.

This information source has, however, limitations mainly due to the fact that they are based on data provided by ISPs and not based on the real state of the network. Firstly, the provided information is often incomplete for various reasons such as confidentiality and administrative overhead. Secondly, as shown in RIPE consistency check reports [139], registry data quality is questionable and often inconsistent as information about a same object in one registry overlaps and sometimes even contradicts information in other registries. Thirdly, due to their inherent nature, these registries are not able to precisely reflect the actual state of routing in the network. For instance, it cannot determine whether portions of the Internet are reachable or not, or whether backup links exist and are used.

These limitations are the reason why current work has tended to focus on other information sources for topology discovery at the AS level. Nevertheless, routing registries still provide a useful source of information when combined with other sources.

5.3.2 BGP Routing Information

As opposed to link-state protocols such as OSPF [54] or IS-IS [140], BGP does not maintain any unified view of the network. Each BGP router chooses its best path for a specific destination which is propagated to its neighbors, leading to an individual view of the network for each router. This view depends on factors such as the choices made by its neighbors, the order in which it receives their announcements, etc. This distributed nature calls for the use of information gathering methods in order to obtain the most complete common view of the topology.

Common BGP information sources are *looking glasses* and *route servers*. A looking glass is a web interface to a BGP router which usually allows BGP data querying and limited use of debugging tools such as `ping` and `traceroute`. A route server is a BGP router offering interactive login access permitting to run most non-privileged router commands. Both are usually made public to help network operators in their debugging tasks, but they can also provide BGP information to properly crafted network discovery tools. A list of accessible looking glasses and route servers is available at [141].

A second source of BGP information is *BGP dumps*. Projects such as RouteViews [142] or RIPE NCC provide collected information from BGP routers around the world. Route collectors are deployed in various locations and peer with BGP routers from multiple ASes. They then periodically save snapshots of their state, known as *table dumps*, along with all routing updates received between the preceding and current snapshot, known as *update traces*. Another way to get BGP information is to use a Zebra router configured to log all BGP update messages. Zebra is an open-source routing daemon [143].

There are several advantages to AS level topology discovery using BGP routing information. First, in the fashion of routing registries, data has been gathered and is available at specific places. There is therefore no need to deploy an infrastructure for exploring the network. Secondly, unlike routing registry data, provided information by BGP corresponds to the actual state of the network, even though it only provides local views of it. Finally, BGP update traces allows dynamic behavior analysis such as backup link detection.

Using BGP routing information has, however, drawbacks. As noticed by Chang et al. [144], BGP does not provide complete information due to missing AS relationships that include both p2c and p2p type relationships. Further, BGP routing information seems to provide a less complete picture of interdomain routing as for example using node-probing, confirmed by Broido and claffy studies [145].

5.4 Inference Techniques

Early research assumed that two ASes were linked if their AS numbers were adjacent in an AS path. Gao and Rexford [146] then made a substantial advance

by noticing c2p links creating so a hierarchy. Gao went on to identify the p2p and s2s relationships [147].

Inferring these relationships is a problem of its own. In her study, Gao [147] first tackles the problem by developing an inference mechanism which extracts information from BGP tables and summarized *valley-free*⁶ property of AS paths. Subramanian et al. [133] formulates AS relationship assignment as an optimization problem, a *type of relationship* (ToR) problem. Battista et al. [148] prove its NP-completeness and present an approximately optimal solution. Gao and Xia [149] evaluate then the accuracy of these algorithms and improve them by introducing techniques on inferring relationships from partial information, in particular the information coming from the BGP community attribute [150]. Dimitropoulos et al. [151] provides improvements to relationships inference. In particular, they provide techniques for inferring s2s from IRRs and heuristics for c2p and p2p relationships.

Chang et al. show that many existing links do not actually appear in BGP [152]. Therefore, they propose to infer the AS topology from Internet's router topology. Broido and claffy [145] reports that the obtained topology differs from the BGP inferred ones in having much denser inter-AS connectivity. It is also richer because it is capable of exposing multiple points of contact between ASes. This is in contrast to BGP table dumps that only provide information on whether two ASes peer or not.

A side issue in inferring AS topologies is to delineate the border of an AS. Indeed, border routers can be made of IP addresses belonging to the AS of interest, to a peer, or to a third party such as an Internet eXchange Point (IXP). Solutions to this issue have been proposed for AS topologies collected by IGMP probing [153] and by `traceroute` [152, 154].

5.5 Limitations

Several works [151, 155–157] demonstrate that AS level topology discovery based on current data collection efforts is limited. Indeed, for instance, Dimitropoulos et al. [151] show that BGP tables missed up to 80% of the ASes adjacencies (mainly p2p relationships). Dimitropoulos et al. [158] suggest to additionally consider BGP updates as the path exploration process may reveal new links between ASes. Others [144] suggest to actually mix the various data source (see Sec. 5.3) instead of considering each source in isolation to others.

Inferring AS level topology from `traceroute` is not exempt from limitations. It naturally comes with all drawbacks inherited from `traceroute` (see Sec. 2.3). In addition, it comes with another limitations called *third-party address*.

A third-party address is the address of a router interface that does not lie in the actual path taken by the packet [52]. Fig. 14 illustrates the problem of third-party address. Remind that, as explained in Sec. 2.1, the source address of the

⁶ After traversing a p2c or a p2p edge, the AS path cannot traverse a c2p or p2p edge. In other words, an AS does not provide transit between any two of its providers or peers.

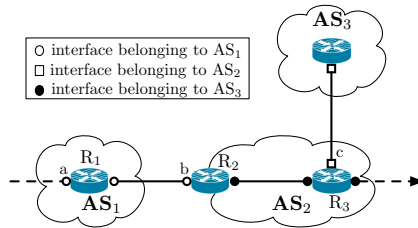


Fig. 14. Third-party address in a traceroute path [53]

`time-exceeded` message generated by a router is the outgoing interface of the reply, not the interface that received the packet generating the reply. On Fig. 14, let us assume that our `traceroute` contains the sequence `a`, `b`, `c` where `a` and `b` are incoming interfaces from router R_1 and R_2 and `c` the interface used by R_3 to send ICMP messages. `c` is a third-party address as it does belong to the actual path traversed by `traceroute` packets.

Third-party addresses have an impact on AS level topology. Indeed, when mapping IP addresses to their AS number, one can generate a false link. On Fig. 14, the third party address `c` generates a false AS link between AS_1 and AS_2 .

Hyun et al. [52] suggest that third-party addresses occur generally at a few hops from the `traceroute` destination (i.e., at the destination edge of the network) and are found mainly in multihomed Stub networks. Marchetta et al. [53] use the IP timestamp option for revealing third-party addresses in a `traceroute`.

6 Conclusion

The Internet is an heterogeneous system made of interconnected entities allowing the communication between machines, from computers to smartphones. In this chapter, we have reviewed how data can be collected for obtaining the Internet topology. In particular, we focused on four levels of the topology: the IP interface, the router, the subnet, and, finally, the AS level. Each level has its own set of inference techniques (active or passive) with their advantages and drawbacks.

Does this chapter mean that everything has been done regarding Internet topology? Surely not. Several challenges are still open. For instance, large-scale distributed measurement infrastructures made of hundreds or thousands of monitors are more and more deployed (see, for instance, the recent RIPE Atlas [159]). Future challenges will concern, for instance, the distribution of gathered data among the measurement points and how to efficiently query this distributed database to provide to the research community or to an application information about the Internet topology.

Challenges are also on network measurement techniques and modeling. For instance, mechanisms for obtaining information about the network dynamics cannot rely on standard probing techniques. Further, current modeling approaches do not take into account network dynamics.

References

1. Pastor-Satorras, R., Vespignani, A.: *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press (2004)
2. Govindan, R., Tangmunarunkit, H.: Heuristics for internet map discovery. In: Proc. IEEE INFOCOM (March 2000)
3. Teixeira, R., Marzullo, K., Savage, S., Voelker, G.: In search of path diversity in ISP networks. In: Proc. ACM SIGCOMM Internet Measurement Conference, IMC (October 2003)
4. Feldman, D., Shavitt, Y., Zilberman, N.: A structural approach for PoP geolocation. *Computer Networks (COMNET)* 56(3), 1029–1040 (2012)
5. Shavitt, Y., Zilberman, N.: Geographical Internet PoP level maps. In: Proc. Traffic Monitoring and Analysis Workshop, TMA (March 2012)
6. Tozal, M.E., Sarac, K.: TraceNET: an Internet topology data collector. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2010)
7. Tozal, M.E., Sarac, K.: Estimating network layer subnet characteristics via statistical sampling. In: Proc. IFIP Networking (May 2012)
8. Gunes, M., Sarac, K.: Inferring subnets in router-level topology collection studies. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2007)
9. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the Internet topology. In: Proc. ACM SIGCOMM (September 1999)
10. Haddadi, H., Iannaccone, G., Moore, A., Mortier, R., Rio, M.: Network topologies: Inference, modeling and generation. *IEEE Communications Surveys and Tutorials* 10(2), 48–69 (2008)
11. Alderson, D., Li, L., Willinger, W., Doyle, J.C.: Understanding Internet topology: Principles, models and validation. *IEEE/ACM Transactions on Networking* 13(6), 1205–1218 (2005)
12. Willinger, W., Alderson, D., Doyle, J.C.: Mathematics and the Internet: a source of enormous confusion and great potential. *Notices of the American Mathematical Society* 56(5), 586–599 (2009)
13. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* 286, 509–512 (1999)
14. Mérindol, P., Donnet, B., Bonaventure, O., Pansiot, J.J.: On the impact of layer-2 on node degree distribution. In: Proc. ACM/USENIX Internet Measurement Conference (IMC (November 2010)
15. Palmer, C.R., Siganos, G., Faloutsos, M., Faloutsos, C., Gibbons, P.B.: The connectivity and fault-tolerance of the Internet topology. In: Proc. Workshop on Network-Related Data Management (May 2001)
16. Radoslavov, P., Tangmunarunkit, H., Yu, H., Govindan, R., Shenker, S., Estrin, D.: On characterizing network topologies and analyzing their impact on protocol design. Technical Report 00-731, Computer Science Department, University of Southern California (February 2000)
17. Jacobson, V., et al.: Traceroute. Man page, UNIX (1989), <ftp://ftp.ee.lbl.gov/traceroute.tar.gz>
18. Sommers, J., Eriksson, B., Barford, P.: On the prevalence and characteristics of MPLS deployments in the open Internet. In: ACM SIGCOMM Internet Measurement Conference (November 2011)
19. Donnet, B., Luckie, M., Mérindol, P., Pansiot, J.J.: Revealing MPLS tunnels obscured by traceroute. *ACM SIGCOMM Computer Communication Review* 42(2), 87–93 (2012)

20. Mérindol, P., Van den Schriek, V., Donnet, B., Bonaventure, O., Pansiot, J.J.: Quantifying ASes multiconnectivity using multicast information. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2009)
21. Gavron, E.: NANOG traceroute, <ftp://ftp.login.com/pub/software/traceroute/>
22. Baker, F.: Requirements for IP version 4 routers. RFC 1812, Internet Engineering Task Force (June 1995)
23. Torren, M.: Tcptraceroute - a traceroute implementation using TCP packets. Man page, UNIX (2001), <http://michael.toren.net/code/tcptraceroute/>
24. Luckie, M., Hyun, Y., Huffaker, B.: Traceroute probe method and forward IP path inference. In: Proc. ACM SIGCOMM Internet Measurement Conference, IMC (October 2008)
25. Mao, Z.M., Johnson, D., Rexford, J., Wang, J., Katz, R.H.: Scalable and accurate identification of AS-level forwarding paths. In: Proc. IEEE INFOCOM (April 2004)
26. Mao, Z.M., Rexford, J., Wang, J., Katz, R.H.: Towards an accurate AS-level traceroute tool. In: Proc. ACM SIGCOMM (August 2003)
27. Zhang, Y., Oliveira, R., Zhang, H., Zhang, L.: Quantifying the Pitfalls of Traceroute in AS Connectivity Inference. In: Krishnamurthy, A., Plattner, B. (eds.) PAM 2010. LNCS, vol. 6032, pp. 91–100. Springer, Heidelberg (2010)
28. Claffy, K., Hyun, Y., Keys, K., Fomenkov, M., Krioukov, D.: Internet mapping: from art to science. In: Proc. IEEE Cybersecurity Applications and Technologies Conference for Homeland Security, CATCH (March 2009)
29. Huffaker, B., Plummer, D., Moore, D.: claffy, k.: Topology discovery by active probing. In: Proc. Symposium on Applications and the Internet, SAINT (January 2002)
30. Luckie, M.: Scamper: a scalable and extensible packet prober for active measurement of the Internet. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2010)
31. Georgatos, F., Gruber, F., Karrenberg, D., Santcroos, M., Susanj, A., Uijterwaal, H., Wilhelm, R.: Providing active measurements as a regular service for ISPs. In: Proc. Passive and Active Measurement Workshop, PAM (April 2001)
32. McGregor, A., Braun, H.W., Brown, J.: The NLANR network analysis infrastructure. IEEE Communications Magazine 38(5), 122–128 (2000)
33. Shavitt, Y., Shir, E.: DIMES: Let the internet measure itself. ACM SIGCOMM Computer Communication Review 35(5), 71–74 (2005), <http://www.netdimes.org>
34. Anderson, D.P., Cobb, J., Korpela, E., Lebofsky, M., Werthimer, D.: SETI@home: An experiment in public-resource computing. Communications of the ACM 45(11), 56–61 (2002), <http://setiathome.berkeley.edu/>
35. Waddington, D.G., Chang, F., Viswanathan, R., Yao, B.: Topology discovery for public IPv6 networks. ACM SIGCOMM Computer Communication Review 33(3), 59–68 (2003)
36. Lang, X., Zhou, G., Gong, C., Han, W.: Dolphin: the measurement system for the next generation Internet. In: Proc. 4th International Conference on Communications, Internet and Information Technology, CIIT (November 2005)
37. Liu, Z., Luo, J., Wang, Q.: Large-scale topology discovery for public IPv6 networks. In: Proc. International Conference on Networking, ICN (April 2008)

38. Madhyastha, H.V., Isdal, T., Piatek, M., Dixon, C., Anderson, T., Krishnamurthy, A., Venkataramani, A.: iPlane: An information plane for distributed services. In: Proc. USENIX Symposium on Operating Systems Design and Implementation, OSDI (November 2006)
39. Sherwood, R., Bender, A., Spring, N.: Discarte: a disjunctive Internet cartographer. In: Proc. ACM SIGCOMM (August 2008)
40. Postel, J.: Internet protocol. RFC 791, Internet Engineering Task Force (September 1981)
41. Fonseca, R., Porter, G., Katz, R., Shenker, S., Stoica, I.: IP options are not an option. Technical Report UCB/EECS-2005-24, University of California, EECS Department (December 2005)
42. Botta, A., de Donato, W., Pescapé, A., Ventre, G.: Discovering topologies at router level: Part II. In: Proc. IEEE Global Telecommunications Conference, GLOBECOM (November 2007)
43. Wide: Gulliver: Distributed active measurement project (2006), <http://gulliver.wide.ad.jp/>
44. Barford, P., Bestavros, A., Byers, J., Crovella, M.: On the marginal utility of network topology measurements. In: Proc. ACM SIGCOMM Internet Measurement Workshop, IMW (November 2001)
45. Guillaume, J.L., Latapy, M.: Relevance of massively distributed explorations of the Internet topology: Simulation results. In: Proc. IEEE INFOCOM (March 2005)
46. Shavitt, Y., Weinsberg, U.: Quantifying the importance of vantage points distribution in Internet topology measurements. In: Proc. IEEE INFOCOM (April 2009)
47. Chen, K., Choffnes, D., Potharaju, R., Chen, Y., Bustamante, F., Pei, D., Zhao, Y.: Where the sidewalk ends: Extending the Internet AS graph using traceroutes from P2P users. In: Proc. ACM SIGCOMM CoNEXT (December 2009)
48. Latapy, M., Magnien, C., Ouédraogo, F.: A radar for the Internet. In: Proc. International Workshop on Analysis of Dynamics Networks, ADN (December 2008)
49. Magnien, C., Ouédraogo, F., Valadon, G., Latapy, M.: Fast dynamics in Internet topology: Preliminary observations and explanations. In: Proc. International Conference on Internet Monitoring and Protection, ICIMP (May 2009)
50. Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., Teixeira, R.: Avoiding traceroute anomalies with Paris traceroute. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (October 2006)
51. Viger, F., Augustin, B., Cuvellier, X., Magnien, C., Friedman, T., Teixeira, R.: Detection, understanding, and prevention of traceroute measurement artifacts. *Computer Networks* 52(5), 998–1018 (2008)
52. Hyun, Y., Broido, A., Claffy, K.: On third-party addresses in traceroute paths. In: Proc. Passive and Active Measurement Workshop, PAM (April 2003)
53. Marchetta, P., de Donato, W., Pescapé, A.: Detecting third-party addresses in traceroute IP paths. In: Proc. ACM SIGCOMM (August 2012) (extended abstract)
54. Moy, J.: OSPF version 2. RFC 2328, Internet Engineering Task Force (April 1998)
55. Callon, R.: Use of OSI IS-IS for routing in TCP/IP and dual environments. RFC 1195, Internet Engineering Task Force (December 1990)

56. Thaler, D., Hopps, C.: Multipath issues in unicast and multicast next-hop selection. RFC 2991, Internet Engineering Task Force (November 2000)
57. Quoitin, B., Uhlig, S., Pelsser, C., Swinnen, L., Bonaventure, O.: Interdomain traffic engineering with BGP. *IEEE Communication Magazine* 41(5), 122–128 (2003)
58. CISCO: How does load balancing work?
http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094820.shtml
59. Juniper: Configuring load-balance per-packet action,
<http://www.juniper.net/techpubs/software/junos/junos70/swconfig70-policy/html/policy-actions-config11.html>
60. Luckie, M., Dhamdhere, A., Claffy, K., Murrel, D.: Measured impact of crooked traceroute. *ACM SIGCOMM Computer Communication Review* 41(1), 15–21 (2011)
61. Spring, N., Mahajan, R., Wetherall, D.: Measuring ISP topologies with Rocketfuel. In: *Proc. ACM SIGCOMM* (August 2002)
62. Augustin, B., Teixeira, R., Friedman, T.: Measuring load-balanced paths in the Internet. In: *Proc. ACM/USENIX Internet Measurement Conference, IMC* (November 2007)
63. Augustin, B., Friedman, T., Teixeira, R.: Measuring multipath routing in the Internet. *IEEE/ACM Transactions on Networking* 19(3), 830–840 (2011)
64. Veitch, D., Augustin, B., Friedman, T., Teixeira, R.: Failure control in multipath route tracing. In: *Proc. IEEE INFOCOM* (April 2009)
65. Donnet, B., Raoult, P., Friedman, T., Crovella, M.: Efficient algorithms for large-scale topology discovery. In: *Proc. ACM SIGMETRICS* (June 2005)
66. Donnet, B., Raoult, P., Friedman, T., Crovella, M.: Deployment of an algorithm for large-scale topology discovery. *IEEE Journal on Selected Areas in Communications (JSAC)*, Sampling the Internet: Techniques and Applications 24(12), 2210–2220 (2006)
67. Spring, N., Wetherall, D., Anderson, T.: Scriptroute: A public Internet measurement facility. In: *Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS)*, pp. 225–238 (March 2003)
68. Donnet, B., Friedman, T., Crovella, M.: Improved Algorithms for Network Topology Discovery. In: Dovrolis, C. (ed.) *PAM 2005*. LNCS, vol. 3431, pp. 149–162. Springer, Heidelberg (2005)
69. Donnet, B., Friedman, T.: Topology discovery using an address prefix based stopping rule. In: *Proc. EUNICE Workshop* (July 2005)
70. Donnet, B., Huffaker, B., Friedman, T., Claffy, K.: Increasing the Coverage of a Cooperative Internet Topology Discovery Algorithm. In: Akyildiz, I.F., Sivakumar, R., Ekici, E., Oliveira, J.C.d., McNair, J. (eds.) *NETWORKING 2007*. LNCS, vol. 4479, pp. 738–748. Springer, Heidelberg (2007)
71. Beverly, R., Berger, A., Xie, G.: Primitives for active Internet topology mapping: Toward high-frequency characterization. In: *Proc. ACM/USENIX Internet Measurement Conference, IMC* (November 2010)
72. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol label switching architecture. RFC 3031, Internet Engineering Task Force (January 2001)
73. Muthukrishnan, K., Malis, A.: A core MPLS IP VPN architecture. RFC 2917, Internet Engineering Task Force (September 2000)
74. Srinivasan, C., Bloomberg, L.P., Viswanathan, A., Nadeau, T.: Multiprotocol label switching (MPLS) traffic engineering (TE) management information base (MIB). RFC 3812, Internet Engineering Task Force (June 2004)

75. Xiao, X., Hannan, A., Bailey, B.: Traffic engineering with MPLS in the Internet. *IEEE Network Magazine* 14(2) (April 2000)
76. Andersson, L., Minei, I., Thomas, T.: LDP specification. RFC 5036, Internet Engineering Task Force (October 2007)
77. Farrel, A., Ayyangar, A., Vasseur, J.P.: Inter-domain MPLS and GMPLS traffic engineering – resource reservation protocol-traffic engineering (RSVP-TE extensions). RFC 5151, Internet Engineering Task Force (February 2008)
78. Bonica, R., Gan, D., Tappan, D., Pignataro, C.: ICMP extensions for multiprotocol label switching. RFC 4950, Internet Engineering Task Force (August 2007)
79. Bonica, R., Gan, D., Tappan, D., Pignataro, C.: Extended ICMP to support multi-part messages. RFC 4884, Internet Engineering Task Force (April 2007)
80. Jacquin, L., Rocca, V., Kaafar, M.A., Schuler, F., Roch, J.L.: IBTrack: An ICMP black holes tracker. In: *Proc. IEEE Global Communications Conference, GLOBECOM* (December 2012)
81. Katz-Bassett, E., Madhyastha, H., Adhikari, V., Scott, C., Sherry, J., van Wesep, P., Krishnamurthy, A., Anderson, T.: Reverse traceroute. In: *Proc. USENIX Symposium on Networked Systems Design and Implementations, NSDI* (June 2010)
82. He, Y., Faloutsos, M., Krishnamurthy, S., Huffaker, B.: On routing asymmetry in the Internet. In: *Proc. IEEE Global Telecommunications Conference, GLOBECOM* (November 2005)
83. Su, Z.S.: A specification of the Internet protocol (IP) timestamp option. RFC 781, Internet Engineering Task Force (May 1981)
84. Morris, R.: A weakness in the 4.2 BSD unix TCP/IP software. Technical Report 117. Bell Labs (February 1985)
85. Bellovin, S.M.: Security problems in the TCP/IP protocol suite. *ACM SIGCOMM Computer Communication Review* 19(2), 32–48 (1989)
86. Ferguson, P., Senie, D.: Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. RFC 2827, Internet Engineering Task Force (May 2000)
87. Baker, F., Savola, P.: Ingress filtering for multihomed networks. RFC 3704, Internet Engineering Task Force (March 2004)
88. Jin, C., Wang, H., Shin, K.: Hop-count filtering: An effective defense against spoofed DoS traffic. In: *Proc. 10th ACM International Conference on Computer and Communications Security, CCS* (October 2003)
89. Leech, M., Taylor, T.: ICMP traceback messages. Internet Draft (work in progress) draft-ietf-itrace-04.txt, Internet Engineering Task Force (February 2003)
90. Snoeren, A.C., Partridge, C., Sancheq, L.A., Joes, C.E., Tchakountio, F., Kent, S.T., Strayer, W.T.: Hash-based IP traceback. In: *Proc. ACM SIGCOMM* (August 2001)
91. Beverly, R., Bauer, S.: The Spoofer projet: Inferring the extent of source address filtering in the Internet. In: *Proc. Steps to Reducing Unwanted Traffic on the Internet Workshop, SRUTI* (July 2005)
92. Yao, B., Chang, R.V., Waddington, F., Topology, D.: inference in the presence of anonymous routers. In: *Proc. IEEE INFOCOM* (April 2003)
93. Gunes, M.H., Sarac, K.: Resolving anonymous routers in the Internet topology measurement studies. In: *Proc. IEEE INFOCOM* (April 2008)
94. IANA: Special-use IPv4 addresses. RFC 3330, Internet Engineering Task Force (September 2002)

95. Cheswick, B., Burch, H., Branigan, S.: Mapping and visualizing the Internet. In: Proc. ACM/USENIX Annual Technical Conference (June 2000)
96. Broido, A., Claffy, K.: Internet topology: Connectivity of IP graphs. In: International Conference on Internet, Performance and Control of Networks, ITCOM (August 2001)
97. Bilir, S., Sarac, K., Korkmaz, T.: Intersection characteristics of end-to-end Internet paths and trees. In: IEEE International Conference on Network Protocols, ICNP (November 2005)
98. Jin, X., Yiu, W.P.K., Chan, S.H.G., Wang, Y.: Network topology inference based on end-to-end measurements. *IEEE Journal on Selected Areas in Communication, Sampling the Internet: Techniques and Applications* 24(12), 2182–2195 (2006)
99. Gunes, M.H., Sarac, K.: Importance of IP alias resolution in sampling Internet topologies. In: Proc. IEEE Global Internet Symposium (May 2007)
100. Braden, R.: Requirements for internet hosts. communication layers. RFC 1122, Internet Engineering Task Force (October 1989)
101. Keys, K.: Iffinder A tool for mapping interfaces to routers, <http://www.caida.org/tools/measurement/iffinder/>
102. Bender, A., Sherwood, R., Spring, N.: Fixing Ally's growing pains with velocity modeling. In: Proc. ACM SIGCOMM Internet Measurement Conference, IMC (October 2008)
103. Keys, K., Hyun, Y., Luckie, M.: claffy, k.: Internet-scale IPv4 alias resolution with MIDAR: System architecture. Technical Report v.0, Cooperative Association for Data Analysis (CAIDA) (May 2011)
104. Kim, S., Harfoush, K.: Efficient estimation of more detailed Internet IP maps. In: Proc. IEEE International Conference on Communications, ICC (June 2007)
105. Zhang, M., Ruan, Y., Pai, V., Rexford, J.: How DNS misnaming distorts Internet topology mapping. In: Proc. USENIX Annual Technical Conference (May/June 2006)
106. Keys, K.: Internet-scale IP alias resolution techniques. *ACM SIGCOMM Computer Communication Review* 40(1), 50–55 (2010)
107. Tozal, M.E., Sarac, K.: Palm tree: an IP alias resolution algorithm with linear probing complexity. *Computer Communications (COMCOM)* 34(5), 658–669 (2011)
108. Fuller, V., Li, T.: Classless inter-domain routing (CIDR): the Internet address assignment and aggregation plan. RFC 4632, Internet Engineering Task Force (August 2006)
109. Sherry, J., Katz-Bassett, E., Pimenova, M., Madhyastha, H.V., Anderson, T., Krishnamurthy, A.: Resolving IP aliases with prespecified timestamps. In: Proc. ACM/USENIX Internet Measurement Conference, IMC (November 2010)
110. Cooperative Association for Internet Data Analysis (CAIDA): Motu dealiasing tool (October 2011), <http://www.caida.org/tools/measurement/motu/>
111. de Donato, W., Marchetta, P., Pescapé, A.: A Hands-on Look at Active Probing Using the IP Prespecified Timestamp Option. In: Taft, N., Ricciato, F. (eds.) PAM 2012. LNCS, vol. 7192, pp. 189–199. Springer, Heidelberg (2012)
112. Garcia-Jimenez, S., Magana, E., Izal, M., Morato, D.: Validity of router responses for IP aliases resolution. In: Proc. IFIP Networking (May 2012)
113. Spring, N., Dontcheva, M., Rodrig, M., Wetherall, D.: How to resolve IP aliases. Technical Report 04-05-04, UW CSE (May 2004)

114. Gunes, M., Sarac, K.: Analytical IP alias resolution. In: Proc. IEEE International Conference on Communications, ICC (June 2006)
115. Gunes, M.H., Sarac, K.: Resolving IP aliases in building traceroute-based Internet maps. *IEEE/ACM Transactions on Networking (ToN)* 17(6), 1738–1751 (2009)
116. Jacobson, V.: Mrinfo (1995), http://cvsweb.netbsd.org/bsdweb.cgi/src/usr.sbin/mrinfo/?only_with_tag=MAIN
117. Deering, S.: Host extensions for IP multicasting. RFC 1112, Internet Engineering Task Force (August 1989)
118. Pusateri, T.: Distance vector multicast routing protocol version 3 (DVMRP). Internet Draft (Work in Progress) draft-ietf-idmr-dvmrp-v3-11, Internet Engineering Task Force (October 2003)
119. Méridol, P., Donnet, B., Pansiot, J.J., Luckie, M., Hyun, Y.: MERLIN: MEasure the Router Level of the INternet. In: Proc. 7th Euro-nf Conference on Next Generation Internet, NGI (June 2011)
120. Marchetta, P., Méridol, P., Donnet, B., Pescapé, A., Pansiot, J.J.: Topology discovery at the router level: a new hybrid tool targeting ISP networks. *IEEE Journal on Selected Areas in Communication, Special Issue on Measurement of Internet Topologies* 29(6), 1776–1787 (2011)
121. Fenner, B., Handley, M., Holbrook, H., Kouvelas, I.: Protocol independent multicast - sparse mode (PIM-SM: Protocol specification). RFC 4601, Internet Engineering Task Force (August 2006)
122. Marchetta, P., Méridol, P., Donnet, B., Pescapé, A., Pansiot, J.J.: Quantifying and mitigating IGMP filtering in topology discovery. In: Proc. IEEE Global Communications Conference, GLOBECOM (December 2012)
123. Mogul, J., Postel, J.: Internet standard subnetting procedure. RFC 950, Internet Engineering Task Force (August 1985)
124. Tozal, M.E., Sarac, K.: Subnet level network topology mapping. In: Proc. IEEE International Performance Computing and Communications Conference, IPCCC (November 2011)
125. Fenner, W.: Internet group management protocol (IGMP), version 2. RFC 2236, Internet Engineering Task Force (November 1997)
126. Hubbard, K., Kusters, M., Conrad, D., Karrenberg, D., Postel, J.: Internet registry IP allocation guidelines. RFC 2050, Internet Engineering Task Force (November 1996)
127. Retana, A., White, R., Fuller, V., McPherson, D.: Using 31-bit prefixes on IPv4 point-to-point links. RFC 3021, Internet Engineering Task Force (December 2000)
128. Vohra, Q., Chen, E.: BGP support for four-octet AS number space. RFC 4893, Internet Engineering Task Force (May 2007)
129. Tangmunarunkit, H., Govindan, R., Estrin, D., Shenker, S.: The impact of routing policy on Internet paths. In: Proc. IEEE INFOCOM (April 2001)
130. Gao, L., Wang, F.: The extent of AS path inflation by routing policies. In: Proc. IEEE Global Internet Symposium (November 2002)
131. Tangmunarunkit, H., Govindan, R., Shenker, S.: Internet path inflation due to policy routing. In: Proc. SPIE International Symposium on Convergence of IT and Communication, ITCOM (August 2001)
132. Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., Jahanian, F.: Internet inter-domain traffic. In: Proc. ACM SIGCOMM (August 2010)

133. Subramanian, L., Agarwal, S., Rexford, J., Katz, R.H.: Characterizing the Internet hierarchy from multiple vantage points. In: Proc. IEEE INFOCOM (June 2002)
134. Zhang, B., Liu, R., Massey, D., Zhang, L.: Collecting the Internet AS-level topology. ACM SIGCOMM Computer Communication Review 35(1), 53–61 (2005)
135. The Internet Society: The regional Internet registry structure, <http://www.isoc.org/briefings/021/>
136. Daigle, L.: WHOIS protocol specification. RFC 3912, Internet Engineering Task Force (September 2004)
137. Network, T.M.: Internet routing database, <ftp://ftp.radb.net/routing.arbiter/radb/dbase/>
138. Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenber, D.: Routing policy specification language (RPSL). RFC 2622, Internet Engineering Task Force (June 1999)
139. RIPE NCC: Routing registry consistency check reports, <http://www.ripe.net/projects/rrcc/>
140. ISO: Intermediate system to intermediate system intra-domain routing exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473) International Standard 10589:2002
141. Kernén, T.: Traceroute organization, <http://www.traceroute.org>
142. University of Oregon: Route views, University of Oregon Route Views project, <http://www.routeviews.org/>
143. Free Software Foundation: Zebra, <http://www.gnu.org/software/zebra/>
144. Changa, H., Govindan, R., Jamina, S., Shenker, S.J., Willinger, W.: Towards capturing representative AS level Internet topologies. Computer Networks (COMNET) 44(6), 737–755 (2004)
145. Broido, A., Claffy, K.: Internet topology: Connectivity of IP graphs. In: Proc. SPIE International Symposium on Convergence of IT and Communication, IT-Com (August 2001)
146. Gao, L., Rexford, J.: Stable Internet routing without global coordination. In: Proc. ACM SIGMETRICS (June 2000)
147. Gao, L.: On inferring autonomous system relationships in the Internet. In: Proc. IEEE Global Internet Symposium (November 2000)
148. Di Battista, G., Patrignani, M., Pizzonia, M.: Computing the types of the relationships between autonomous systems. In: Proc. IEEE INFOCOM (April 2003)
149. Xia, J., Gao, L.: On the evaluation of AS relationship inferences. In: Proc. IEEE Global Communications Conference (GLOBECOM (November 2004)
150. Quoitin, B., Bonaventure, O.: A survey of the utilization of the BGP community attribute. Internet Draft draft-quoitin-bgp-comm-survey-00, Internet Engineering Task Force (February 2002) (work in progress)
151. Dimitropoulos, X., Krioukov, D., Fomenkov, M., Huffaker, B., Hyun, Y., Claffy, K., Riley, G.: AS relationships: Inference and validation. ACM SIGCOMM Computer Communication Review 37(1), 29–40 (2007)
152. Chang, H., Jamin, S., Willinger, W.: Inferring AS-level Internet topology from router-level path traces. In: Proc. SPIE International Symposium on Convergence of IT and Communication, ITCom (August 2001)
153. Pansiot, J.-J., Mérindol, P., Donnet, B., Bonaventure, O.: Extracting Intra-domain Topology from `mrinfo` Probing. In: Krishnamurthy, A., Plattner, B. (eds.) PAM 2010. LNCS, vol. 6032, pp. 81–90. Springer, Heidelberg (2010)

154. Huffaker, B., Dhamdhere, A., Fomenkov, M., Claffy, K.: Toward Topology Dualism: Improving the Accuracy of AS Annotations for Routers. In: Krishnamurthy, A., Plattner, B. (eds.) PAM 2010. LNCS, vol. 6032, pp. 101–110. Springer, Heidelberg (2010)
155. Oliveira, R., Pei, D., Willinger, W., Zhang, B., Zhang, L.: In search of the elusive ground truth: The Internet’s AS-level connectivity structure. In: ACM SIGMETRICS (June 2008)
156. Oliveira, R., Pei, D., Willinger, W., Zhang, B., Zhang, L.: The (in)completeness of the observed Internet AS-level structure. *IEEE/ACM Transactions on Networking* 18(1), 109–122 (2010)
157. Willinger, W., Maennel, O., Perouli, D., Bush, R.: 10 lessons from 10 years of measuring and modeling the Internet’s autonomous systems. *IEEE Journal on Selected Areas in Communications (JSAC)* 29(9), 1810–1821 (2011)
158. Dimitropoulos, X.A., Krioukov, D.V., Riley, G.F.: Revisiting Internet AS-Level Topology Discovery. In: Dovrolis, C. (ed.) PAM 2005. LNCS, vol. 3431, pp. 177–188. Springer, Heidelberg (2005)
159. RIPE NCC: Atlas (2010), <https://atlas.ripe.net/>

Internet PoP Level Maps

Yuval Shavitt and Noa Zilberman

School of Electrical Engineering, Tel-Aviv University, Israel
{shavitt,noa}@eng.tau.ac.il

Abstract. Inferring the Internet Point of Presence (PoP) level maps is gaining interest due to its importance to many areas, e.g., for tracking and studying properties of the Internet. In this chapter we survey research towards the generation of PoP level maps. The chapter introduces different approaches to automatically classify IP addresses to PoPs and discusses their strengths and weaknesses. Special attention is devoted to the challenge of validating the generated PoP maps in the absence of ground truth. The chapter next describes general IP geolocation techniques, points out weaknesses in geolocation databases, as well as, in constraint-based approaches, and concentrates on PoPs geolocation techniques, discussing validation and lack of ground truth availability. The third part of the chapter describes how to generate maps with PoP-to-PoP connectivity and analyzes some of their properties. At the end of the chapter, some applications of PoP level maps, such as Internet distance maps, evolution models and homeland security are introduced and discussed.

1 Introduction

The study of the Internet topology attracted a great deal of work over the years. A good survey of these efforts can be found in the "Internet topology discovery" chapter of this book, as well as in an earlier survey by Donnet and Friedman [9]. Internet topology maps are used for a vast number of applications, such as building models of the Internet [37], studying the robustness of the network [10], network management [47] and improving routing protocols design [40]. There are several levels Internet maps are presented at, each level of abstraction is suitable for studying different aspects of the network. The most detailed level is the IP level, which represents separately each and every entity connected to the network. Many projects map the Internet at the IP level, such as Skitter [23], RIPE NCC's Test Traffic Measurement [14], iPlane [31], DIMES [8], Ark [24], and more. This level is far too detailed to suit practical purposes, and the large number of entities makes it very hard to handle. One level above the IP level is the router level, aggregating multiple IP interface addresses to a router, using alias resolution, as done by projects such as Mercator [15], MIDAR [27], Ally [49], and RadarGun [4]. While being less detailed than the IP level, this level of aggregation is still highly detailed and difficult to handle. The most coarse level is the Autonomous System (AS) level. It is most commonly used

to draw Internet maps, as it is relatively small (tens of thousands of ASes) and therefore relatively easy to handle: there is only one node for every AS, and may have only one edge between every pair of ASes. There are different methods to discover the Internet's AS-level topology, from using traceroutes, as done in Ark, iPlane and DIMES, through BGP announcements, as done by Routeviews [52] to Internet Routing Registries (IRR) [34]. One limitation of using AS information for Internet mapping is that AS sizes may differ by orders of magnitude. While a large AS can span an entire continent, and a small one can serve a small community, yet both seem identical at the AS level map.

An interim level between the AS and the router graphs is the PoP level. Service providers tend to place multiple routers in a single location called a Point of Presence (PoP), which serves a certain geographical area. A PoP is defined as a group of routers which belong to the same AS and are physically located at the same building or campus.

Figure 1 demonstrates the Internet aggregation levels. The figure presents for clarity only the AS, PoP, and router levels. Every AS, marked by a large circle, is made of a network of routers, marked by small light gray circles. The routers may be part of a PoP (colored dark gray), or reside outside of a PoP. A router which is not part of a PoP will still be connected to other routers, eventually connecting to a PoP. The points of presence are connected to other PoPs within the same AS as well as to PoPs outside their AS, thus creating AS level connectivity.

The technological nature of PoPs varies between service providers as well as within the same network. Some PoPs operate entirely on the IP level, while other PoPs employ MPLS and VPLS switching. In many cases, service provider mix switching and routing within the same PoP, combining both MPLS and IP. In more rare cases, in Optical Transport Networks, the PoP may only serve as a channel based cross connect. A good example of this mix is shown in CenturyLink's network [5]: In some cities, such as Atlanta, Los Angeles, and New York City, both IP and MPLS/VPLS are used. In other cities, such as Sacramento, Duluth, and Cambridge, MA, there is an IP PoP, while in cities such as New Orleans, San Antonio, and San Diego only MPLS/VPLS is used. Additional examples can be found in the TeliaSonera network map [51] and XO network map [54]¹. Service provider also tend to distinguish between different types of PoPs, often referring to the hierarchy in the network, e.g., access or backbone PoP [5] or to the area it covers, e.g., a metro PoP [54]. A declining trend is to refer to PoPs by their capacity, such as GigaPoP [25] or TeraPoP² [39].

When studying the entire network, and not only specific ISPs, PoP maps give a better level of aggregation than router level maps with a minimal loss of information. PoP level graphs provide the ability to examine the size of each AS network by the number of physical co-locations and their connectivity instead of by the number of its routers and IP links. Points of presence can be annotated with geographical location, as well as information about the size of the PoP.

¹ This information was also confirmed with a large networking equipment provider.

² As called by Qwest, before Qwest was acquired by CenturyLink.

PoP maps can also preserve routing information by annotating links connecting Pops that belong to different ASes with the type of relationship (ToR). Thus, using PoP level graphs it is possible to detect important nodes of the network and understand network dynamics as well as many more applications.

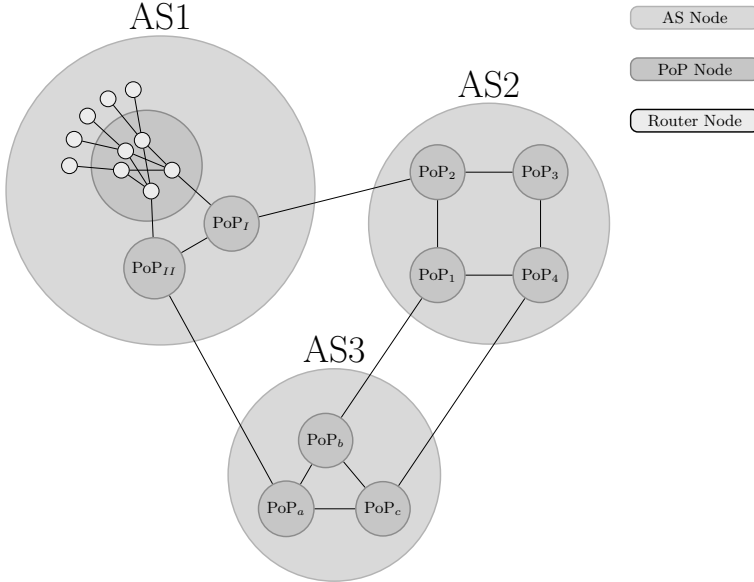


Fig. 1. The Internet's Levels of Aggregation

This chapter surveys the study of Internet PoP level maps, providing an overview of all works done so far in this field. The chapter is organized as follows: Section 2 discusses classification of IP addresses into PoPs and surveys existing works in this field. Section 3 describes some methods for assigning a location to points of presence. The generation of PoP-level connectivity maps is presented in Section 4 and some analysis of the maps is provided. The validation efforts of PoP level maps is surveyed in Section 5. In Section 6 we discuss applications of the PoP level graphs by various disciplines. Last, section 7 concludes this chapter.

2 IP Classification into PoPs

The first attempts to explore the PoP level graph were done by Andersen *et al.* [3] and Spring *et al.* [49]. Spring *et al.* [49] tried to infer ISP topologies both on the router and the PoP level. The focus of their contribution was in alias resolution and router identification based on in-order IP identifiers and introducing Rocketfuel, their mapping engine. The PoP resolution was entirely DNS based. To this end, they inferred ISP naming convention. For example, s1-bb11-nyc-3.0.sprintlink.net is indicated to be a Sprint backbone (bb11) router in

New York City (nyc). The naming convention was deduced from the list of router names they gathered during the alias resolution and router identification stage with some city names taken from [36]. For routers with no DNS names or where the names lacked location information, the locations of neighbor routers were used. The generated PoP map did not distinguish between backbone network nodes, data centers, or private peering points.

Ten ISPs were tested by Spring *et al.* and the number of PoPs discovered per ISP ranged from 11 (AS4755, VSNL India) to 122 (AS2914, Verio US). The PoPs' analysis showed that the designs of PoPs were relatively similar: generic PoPs are built from a few routers connected in a mesh while in large PoPs the access routers connect one or more routers from a neighboring domain and to two backbone routers for redundancy. The backbone routers connect both to routers within the same PoP as well as to routers in other PoPs that connect to the ISP's backbone. The result showed that small PoPs had for redundancy two backbone PoPs, but in large PoPs with 20 routers or more, the number of backbone routers varied significantly, from two to thirty.

Andersen *et al.* [3] used passive monitoring of BGP messages to cluster IP prefixes into PoPs: In the preprocessing stage, BGP messages are grouped into time intervals of I seconds and massive updates due to session resets are filtered. The clustering stage is based on a distance metric, which is a function that determines how closely two items are. The distance metric used is the correlation coefficient between every pair of BGP update vectors. $u_p^{(t)}$ denotes the update vector for each prefix p :

$$u_p^{(t)} = \begin{cases} 1 & \text{if } p \text{ updated during interval } t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$C(p1, p2)$ is the correlation coefficient between two prefixes, with $\overline{u_p}$ being the average of $u_p^{(t)}$ and σ_p its variance.

$$C(p1, p2) = \frac{\frac{1}{n} \sum_{t=1}^n (u_{p1}^{(t)} - \overline{u_{p1}})(u_{p2}^{(t)} - \overline{u_{p2}})}{\sqrt{\sigma_{p1}^2} \sqrt{\sigma_{p2}^2}} \quad (2)$$

A Single-linkage clustering algorithm [57] is applied for grouping prefixes. Using the distance metric presented by Equation 2, each pairwise distance between two prefixes is computed and prefixes with time window of 30 seconds are grouped.

Andersen *et al.* used BGP updates from two upstream feeds: a commercial feed via Genuity (AS 1), and an Internet2 feed via the Northeast Exchange (AS 10578). Due to their configuration, only the best route to every prefix was recorded, thus some paths were omitted from their dataset. The clustering was conducted on 2338 prefixes announced by UUNET (AS 701) and 1310 prefixes announced by AT&T (AS7018) and ended up with 6 clusters in UUNET and 5 clusters in AT&T, with the number of clusters strongly dependent on the number of pairwise comparisons during the clustering phase. The analysis observed the effect of the number of matches on the number of clusters and their accuracy.

The validation was conducted based on three methods: IP address similarity (the number of IP addresses that separate two prefixes), Ratio of shared to unshared traceroute path length (in hops) and DNS-based PoP comparison. The last means that they extracted a router location from the ISP’s naming convention, managing to assign 97% of UUNET hops and somewhat less for AT&T. Their results showed that correlation-based clustering grouped the UUNET prefixes into about 1200 clusters while with over 95% PoP-level accuracy as well as 900 clusters in AT&T with 97% accuracy. The accuracy is defined as a match between the naming conventions. The concluding observation is that clusters that are announced and withdrawn together tend to be located at the same PoP.

The iPlane project [32,31] generates PoP level maps based on the Rocketfuel’s approach, with several improvements: First, they determine the DNS names assigned to network interfaces, using two data sources: Rocketfuel’s `undns` utility [49] and data from the Sarangworld project [1]. DNS alone is not enough, as some interfaces have no DNS names, others have no rules to infer their DNS name and for some interfaces may be misnamed, thus incorrect locations can be inferred [56]. For the last, interfaced are probed using ICMP ECHO packets and interfaces where the RTT is smaller than expected are filtered. The main new contribution in this work is an algorithm that clusters router interfaces based on their responses when probed from a large number of vantage points. iPlane estimates the number of hops on the reverse path back from a router to the vantage point, guessing the initial TTL value used by the router. The assumption is that routers in the same AS and geographically co-located take the same reverse path back to the vantage point from which they were probed, while routers that are not co-located will not display similar reverse path. iPlane detects about 135K PoPs, about 56K of them in singleton clusters, meaning a single router in a cluster. We discuss further the iPlane project in Section 4.

The PoP extraction algorithm proposed by DIMES [12] is based on the fact that in most cases [16,42] the PoP consists of two or more backbone/core routers and a number of client/access routers. The client/access routers are connected redundantly to more than one core router, while core routers are connected to the core network of the ISP. The algorithm takes a structural approach and looks for bi-partite subgraphs³ with certain weight constraints in the IP interface graph of an AS; no aliasing to routers is needed. The bi-partites serve as cores of the PoPs and are extended with other nearby interfaces.

The algorithm works on the Interface graph of each ISP separately. It starts by removing all edges with delay higher than PD_{max_th} , PoP maximal diameter threshold, and edges with number of measurements below PM_{min_th} , the PoP measurements threshold. As a result the ISP interface graph is partitioned to several components, each is a candidate to become one or more PoPs. Next, the algorithm looks at the bi-partites in each component and uses the rich connectivity there between the sources (parents) and destinations (children) to check for node colocation based on link delays between the groups. If *parent* and *child*

³ A bipartite graph is a graph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V .

groups are connected, then the weighted distance between the groups is calculated (If they are connected, by definition more than one edge connects the two groups); if it is smaller than a certain threshold the pair of groups is declared as part of the same PoP. Last, a unification of loosely connected parts of the PoP is conducted. For this end, the algorithm looks for connected components (PoP candidates) that are connected by links whose median distance is very short (below $PD_{max.th}$).

The number of PoPs discovered by DIMES is in the range of 4000 to 5000.

Yoshida *et al.* [55] mapped PoP-paths in Japan using thirteen dedicated measurement nodes and measuring the delay between these nodes. They tried to map the core network delay, derived from the end-to-end delays and access delays and their corresponding PoP level paths, using a set of delay equations:

$$delay(src, dst) = ad_{src} + ad_{dst} + \sum_{p,q \in N} x_{p,q} \times cd_{p,q} + E_{src,dst} \quad (3)$$

In the equation, N denotes a set of candidate PoP locations of a measured ISP; p and q satisfy $p, q \in N$; ad_{src} and ad_{dst} denote the access delay at the source and the destination; $cd_{p,q}$ denotes a core delay between p and q ; $E_{src,dst}$ is the measurement error of the delay; $x_{p,q} = 1$ if a direct path between p and q exists and the path is used to connect between src and dst , otherwise $x_{p,q} = 0$. $delay(src, dst)$, ad_{src} and ad_{dst} are measurable through end-to-end measurements and $cd_{p,q}$ can be derived leveraging the distance between p and q . To solve the equation, several restrictions are applied. One of the assumptions used is that the network connections are deployed along other infrastructure services, such as railroads and expressways.

The work distinguished between five types of networks, differing by the way the backbone routers are structured and by the way layer two is used. For example, is layer three being used in every location in the network, or are layer three routers being used only in highly populated cities.

A different approach to PoP level maps is presented by Rasti *et al.* [41]. They term an eyeball AS as an individual Autonomous Systems that directly provides service to end-users and use the eyeball ASes to estimate the PoP-level footprint. The basic assumption is that each AS must have a PoP in areas it has a high concentration of customers. Therefore, the AS eyeball offers a view of that AS's PoP-level infrastructure, referred to as PoP-level footprint. The algorithm begins by gathering a large number of end-user IP addresses, collected by crawling P2P applications. The users are then mapped to cities using geolocation services (discussed in Section 3) and are grouped to ASes based on Routeview's BGP tables [52]. Given the locations of the users, the geographical regions where the AS offers service to end-users is inferred using KDE (Kernel Density Estimation). To extract the PoP footprint, local maxima $D(i)$ are detected in the density function, with the highest peak denoted by D_{max} . PoPs are indicated by any peak $D(i)$ that is within a given range from D_{max} , meaning $D(i) > \alpha \times D_{max}$, with α set to 0.01. The work focused on 672 ASes and found an average of 13.6 PoPs per AS when using 40km range as the kernel function bandwidth.

To conclude, there are several different approaches to the classification of IP addresses into PoPs. Yet, grouping the IP addresses into PoPs is just the first stage of generating PoP level maps, as we discuss in the following sections.

3 Geolocation of PoPs

An important feature of PoP level maps is the ability to assign a geographical location to PoPs. The assignment is done using geolocation mapping services, providing longitude and latitude or a city and a country per IP address. Geolocation mapping services can be divided to several groups. For mobile devices, GPS is the most common approach to locate a device. A second group of geolocation mapping services is geolocation databases, holding a table mapping every IP address to its geographical location. Geolocation databases range from free services to services that cost tens of thousands of dollars a year. The most basic services use DNS resolution as the basis for the database [49], while others use proprietary means such as random forest classifier rules, hand-labeled hostnames [2], user's information provided by partners [7], and more.

Another group of geolocation mapping services is based on network measurements. IP2Geo [36] was one of the first to suggest a measurement-based approach to approximate the geographical distance of network hosts. A more mature approach is constraint based geolocation [19], using several delay constraints to infer the location of a network host by a triangulation-like method. Later works, such as Octant [53] used a geometric approach to localize nodes within a 22 miles radius. Katz-Bassett *et al.* [26] suggested topology based geolocation using link delay to improve the location of nodes. Yoshida *et al.* [55] used end-to-end communication delay measurements to infer PoP level topology between thirteen cities in Japan. Eriksson *et al.* [11] applied a learning based approach to improve geolocation. They reduced IP geolocation to a machine learning classification problem and used Naive Bayes framework to increase geolocation accuracy.

One online geolocation service that allows querying specific IP addresses is Spotter, which is based on a work by Laki *et al.* [29]. Spotter uses a probabilistic geolocation approach, which is based on a statistical analysis of the relationship between network delay and geographic distance. To approximate the location of a target, Spotter measures propagation delays from landmarks to the target, and then converts the delays into geographic distances based on a delay-distance model. The resulting set of distance constraints is used to determine the targets estimated location with a triangulation-like method.

Not many works have focused on the accuracy of geolocation databases, but those who did showed them to be inaccurate: In 2008, Siwpersad *et al.* [48] examined the accuracy of Maxmind [33] and IP2Location [20]. They assessed their resolution and confidence area and concluded that their resolution is too coarse and that active measurements provide a more accurate alternative. Gueye *et al.* [17] investigated the imprecision of relying on the location of blocks of IP addresses to locate Internet hosts and concluded that geolocation information coming from exhaustive tabulation may contain an implicit imprecision. Muir

and Oorschot [35] conducted a survey of geolocation techniques used by geolocation databases and examined means for evasion/circumvention from a security standpoint. Shavitt and Zilberman [45] studied extensively seven geolocation services both on IP and PoP level. Their results show that the information in the databases may be largely biased at the ISP level: Using a small ground truth database provided by CAIDA of 25K addresses and described in [22], they found that some of the databases place all the IP addresses of a certain ISP in a single location, typically the ISP headquarters' city; while for other ISPs correct location is provided. Additionally, correlation was found between databases: some databases, such as Maxmind [33] and IP2Location [20] have an extremely small median distance between an IP address' geolocations, below 10km, while for other databases, such as GeoBytes [13] and HostIP.info [21] the median distance may be above 500km. The differences between databases may be in the range of countries: in one example case, a 10-nodes PoP was located by some databases in Singapore, by others in Australia, while two more databases pointed to China and Afghanistan, as shown in Figure 2. Constraint based approaches are many times no better than geolocation databases. They inherently have an inaccuracy in the range of tens to hundreds of kilometers [45], and strongly depend on the location of the vantage points. A non optimal location of vantage points may lead to an error in the range of hundreds of kilometers, and more. Poese *et al.* [38] studied five databases and showed that while on the country-level they are rather accurate, the databases are highly biased towards a few popular countries. Using ground truth information from one large European ISP and using DNS names as clues for two large other major ISPs, Poese *et al.* showed that the evaluated databases performed poorly on those ISPs.

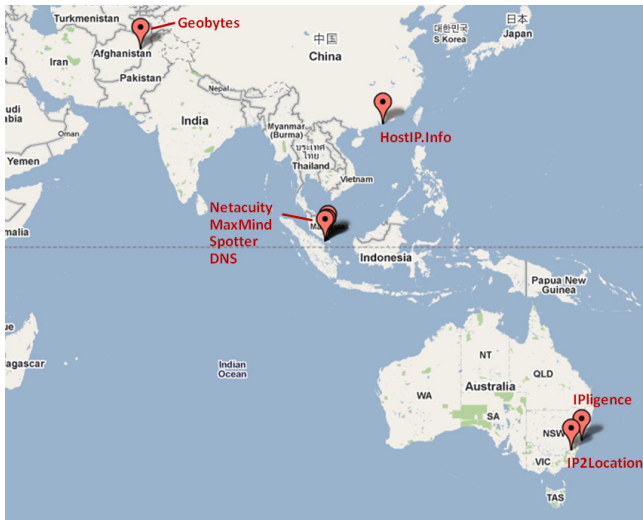


Fig. 2. Mismatch Between Databases - An Example

Most of the PoP extraction algorithms described in Section 2 use a crude method of geolocation as the basis for their geolocation: DNS names. This is an easy to use method, leveraging the fact that the router’s location is often written in the router’s name used by the ISP. However, DNS suffers from several problems: many interfaces do not have a DNS name assigned to them, and incorrect locations are inferred when interfaces are misnamed [56]. In addition, rules for inferring the locations of all DNS names do not exist, and require some manual adjustments. DIMES’ maps take an approach based on geolocation databases: it uses the geographic location of each of the IPs included in a PoP, as denoted by at least three geolocation databases (typically more) and take the median location. A range of error, indicating the radius within 50% of the IP location votes reside, is assigned per PoP and the location of a PoP is further refined based on these locations alone. As all the PoP IP addresses should be located within the same campus, the location confidence of a PoP is significantly higher than the confidence that can be gained from locating each of its IP addresses separately. An example of a PoP level map generated by this method is given in Figure 3. Note that PoPs that appear in mid ocean are actually located on islands. Rasti *et al.* [41] use two geolocation databases: MaxMind GeoIP City [33] as a main source and IP2Location [20] to corroborate MaxMind’s information, discarding IP addresses missing city location in one of the databases or where there is more than 100km deviation between the two sources. This approach is somewhat inaccurate, as Shavitt and Zilberman [45] have shown that these two databases are the most similar out of seven tested databases, with a median of five kilometer distance between IP addresses locations within them. This means that an error in Maxmind database is highly likely to appear in IP2Location database as well.

One way to improve the location provided by geolocation databases is to use the PoP level graph itself, starting at PoPs with a known location (such as universities) or a location with high level of confidence and crawling the



Fig. 3. DIMES PoP Level Map, 2010

graph to improve the location of neighboring nodes: The algorithm starts by identifying and marking the PoPs whose location is certain. The algorithm then discovers PoPs that are located in the same place as the marked PoPs, based on the PoP-level link delay. Following, it attempts to find the optimal location of non-marked PoPs based on the ratio of PoP-link delay to PoP distance from marked PoPs. First, all the possible locations for all the PoP IP addresses from all databases are examined to find the one that has the best ratio. If the best location is not satisfying, multilateration is used. The algorithm then iterates and tries to improve the location of unmarked PoPs using the location of newly marked PoPs.

The algorithm can then be extended for IP geolocation. It was shown that 80% of the IP addresses were within 1mS and two hops from a PoP and for the rest of the IPs multilateration can be applied from the nearest PoPs. The algorithm was corroborated by a large ISP where geolocation databases placed all of its IP addresses in a single location. The algorithm correctly distributed the ISP's IP addresses to near PoP locations around the globe.

Assigning a geographical location to PoPs is therefore a difficult task which is hard to validate without ground truth information.

4 PoP-Level Connectivity

The connectivity between PoPs is an important part of PoP level maps. DIMES generates PoPs connectivity graph using unidirectional links. They define a link L_{SD} as the aggregation of all unidirectional edges originating from an IP address included in a PoP S and arriving at an IP address included in a PoP D . Each of the IP level links has an estimate of the median delay measured along it, with the median calculated on the minimal delay of up to four consecutive measurements. Every such four measurements comprise a basic DIMES operation. All measured values are roundtrip delays [12]. A Link has the following properties:

- Source and Destination PoP nodes.
- The number of edges aggregated within the link.
- Minimal and Maximal median delays of all IP edges that are part of the PoP level link.
- Mean and standard deviation of all edges median delays.
- Weighted delay of all edges median delays. The edge's weight is the number of times it was measured.
- The geographical distance between source and destination PoP, calculated based on the PoPs geolocation.

A weighted delay of a link is used to mitigate the effect of an edge with a single measurement on the overall link delay estimation, where a link is otherwise measured tens of times through other edges. iPlane uses the inter-cluster connectivity to generate PoP level connectivity, with a similar definition of PoP level links, only using bidirectional links. The delay measured on links is not very different than in DIMES': For every inter-PoP link, iPlane considers all the corresponding

inter-IP links that are measured in traceroutes. From every traceroute in which any such inter-IP link is observed, obtain one latency sample for the link as the difference in RTTs to the either end of the link and drop all latency samples that are below 0. Compute the latency for the inter-PoP link as the median of all the remaining samples for it. If there are no samples left after ignoring all the negative latency samples, the latency of the link is indicated as -9999 (about 6% of the links).

Using a dataset comprised of 478 million traceroutes conducted in weeks 42 and 43 (late October) of 2010, measured by 1308 DIMES agents and 242 iPlane vantage nodes and applying DIMES' algorithm to it result in a PoP level map that contains 4750 PoPs, 82722 IP addresses within the PoPs and 102620 PoP level links [46]. The links are an aggregation of 1.98M IP level edges. All the PoPs have outgoing links, with only 2 PoPs having only incoming links and one PoP with no PoP level links (only IP-level). As a full PoP level map is too detailed to display, a partial map is shown in Figure 4. The figure demonstrates the connectivity between randomly selected 430 ASes on the PoP level.

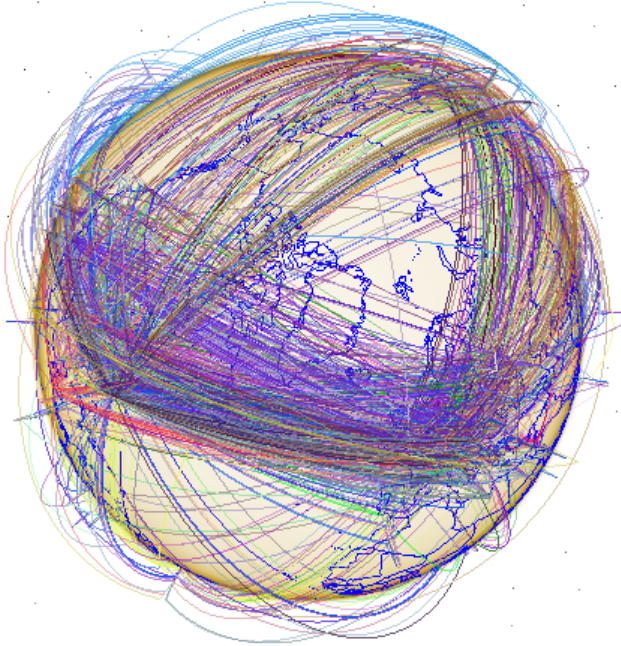


Fig. 4. An Internet PoP Level Connectivity Map - A Partial DIMES Map of Week 42, 2010

Most of the IP edges that are aggregated into links are unidirectional: 96.6%. This is a characteristic of active measurements: vantage points are limited in number and location, thus most of the edges can be measured only one way. However, at the PoP level, 18.8% of the links are bi-directional: six times more than the bi-directional edges. This demonstrates one of the advantages of using a PoP level

map, as it provides a more comprehensive view of the networks' connectivity without additional resources. The average number of edges within a unidirectional link is 6.9, and the average number of edges within a bidirectional link is 72.9. This is not surprising, as it is likely that most of the bidirectional links will connect major PoPs, within the Internet's core and thus be easily detected.

An additional view of edges aggregation into links is given by Figure 5. The X-axis shows the number of edges aggregated into a link, while the Y-Axis is the number of PoP-level links. The graph shows a Zipf's law relation between the two, as 81.5% of the links aggregate ten edges or less, and less than 2.5% aggregate 100 edges or more. The large number of edges per link is explained by the fact that a measured edge is not a point-to-point physical connection: Take two routers, A & B, connected by a single fiber; If one of the routers has 48 ports, and one measures through each and every port, he will detect 48 edges between the two routers (incoming port i on router A and the single connected incoming port of router B).

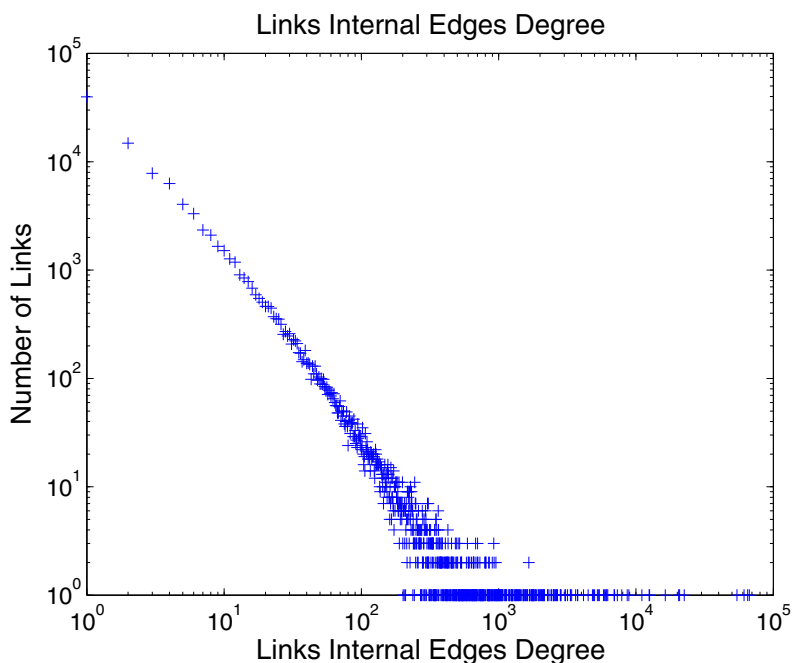


Fig. 5. Number of Edges within a Link vs. Number of PoP Level Links in DIMES Dataset

The number of links per PoP also behaves according to Zipf's law, as shown in Figure 6. The figure shows the total number of links per PoP, the number of outgoing links (source PoP) and the number of incoming links (Destination port). The connectivity between PoPs is very rich: only twenty two PoPs have

one or two links to other PoPs, while 70% of the PoPs have ten or more links to other PoPs.

Many of the links are between PoPs that are co-located, which we define as links with a minimal delay of 1mS or less, and over 90% of the PoPs have such links. Almost all the PoPs (over 97%) are connected to PoPs outside their AS.

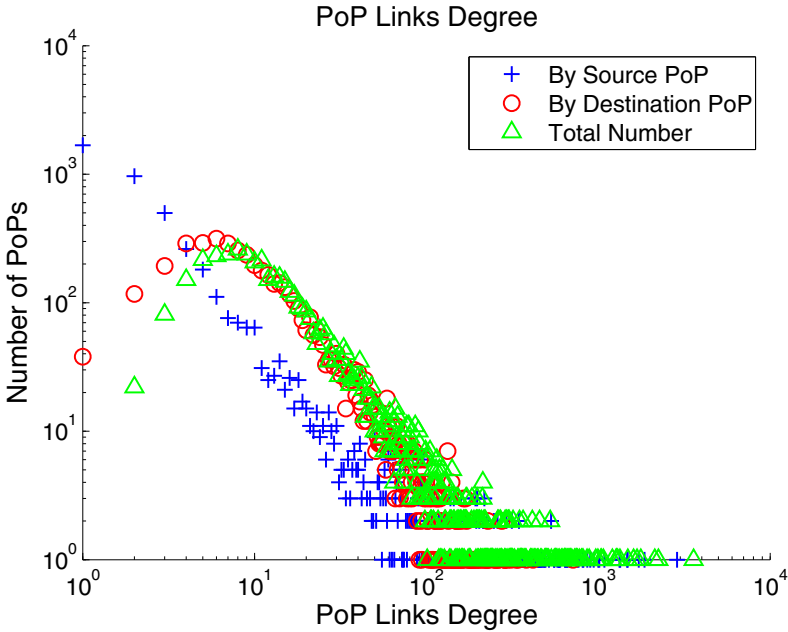


Fig. 6. Number of Links per PoP vs. Number of PoPs in DIMES Dataset

Figure 7 shows the minimum, weighted average and maximal delay per link, plotted on a log-log scale with the delay (X-scale) measured in milliseconds. The solid black line shows the cumulative number of measurements up to a given link delay. We omit from this plot links that include only a single edge, which distort the picture as their minimal, weighted and maximal delay are identical. An interesting attribute of this plot is that all three plotted delay parameters behave similarly and are closely grouped. As all the links are an aggregation of multiple edges, this indicates the similarity in the delay measured on different edges. One can also see that most of the measurements represent a delay of 200ms or less, and that the extreme cases are rare (see the cumulative measurement line). In almost all the cases where a minimal delay of 1sec or more are measured, this is a link that is made of a single edge. The same logic applies also for links with a small maximal delay, meaning the maximal delay was defined by only one or two measured edges. Here, however, a small maximal delay may also indicate co-located PoPs.

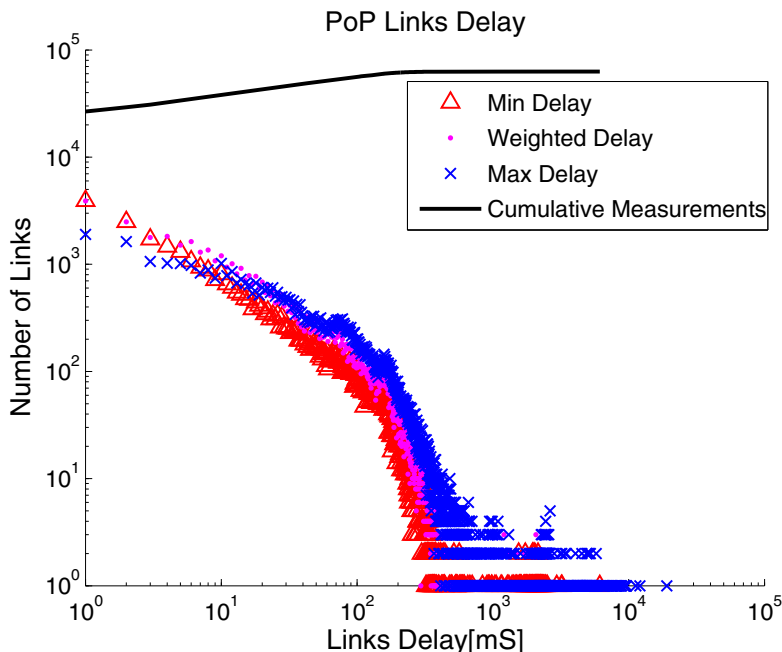


Fig. 7. Links Delay vs. Number of Links in DIMES Dataset

Traceroute measurements are known to introduce delay errors [18,44]. The errors tend to be of an additive nature, though sometimes a measured single-edge delay may be lower than its physical delay, due to an additive delay of the previous edge in the measured traceroute. This phenomenon is demonstrated by Figure 8: The X-Axis of the figure shows the estimated minimal link delay (in milliseconds), and the Y-Axis shows the spread of edge delay measurements. The figure focuses on the interesting range of delays, up to 500mS link delay and one second edge delay. A few measurements exist outside these boundaries, but their contribution to this discussion is small. Figure 8 clearly demonstrates the effect of a single edge measurement error: some links have a minimum delay of zero yet some of their measurements reach one second. Thus the aggregation of multiple edges into PoP level links significantly cleans noise from the collected data.

The Internet Topology Zoo [28] is a project that stores a large number of PoP level maps obtained from service providers' websites. The project provides PoP connectivity maps in GML format, converted from the original image file. The maps are annotated, when possible, with the following information:

- link types or speeds.
- longitudes and latitudes of nodes obtained through geocoding of PoP locations.
- a URL showing where the data was obtained.

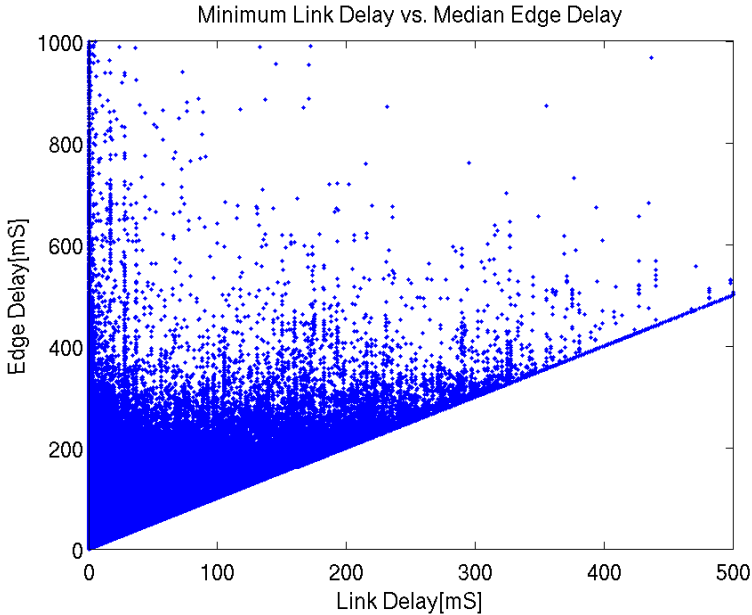


Fig. 8. Link Delay vs. Edge Delay in DIMES Dataset

- the date-of-record, i.e., the date that the map was representative of the network.
- the date the network map was obtained.
- a classification of the type of network.

In the Topology Zoo analysis, 59 research and education PoP-level networks and 82 commercial networks were studied. Most of the networks under study were backbone networks, and operated on country level or lower. Half of the networks had more than 21 PoPs, and 10% had 51 PoPs or more. The node degree observed was very low, ranging in different networks from 1.66 to 4.5, however this was only Intra-AS connectivity and was also biased as many of the maps were partial. For example, Sprint's network appeared to contain only 11 PoPs, and in AT&T only the MPLS network was present.

5 PoP Maps Validation

An important question when examining PoP level maps is how the map was validated. Accuracy is the most important validation evaluation aspect, and it entails multiple facets.

- How accurate is the classification of IP addresses to PoPs?
- How accurate is the assignment of PoPs to geographic locations?
- How accurate is the inference of PoP level links and their delay?

In addition, one may also want to evaluate the coverage of PoPs, meaning how many of the actual ISP's PoPs are covered by the extracted PoPs map, and how many IPs of a PoP are assigned to it. The effort required to validate PoP level maps is thus considerably high.

Spring *et al.* [49] verified completeness with the help of three ISPs. The ISPs verified that no PoPs or inter-PoP links were missing. However, in two of the cases there were spurious links. In addition, some access PoPs were missing. Further validation was conducted on the router level, both for completeness, impact of measurement reductions and alias resolution. The alias resolution, used for PoPs detection, failed for about 10% of the IP addresses, and in Sprint network, 63 out of 303 routers were resolved incorrectly.

Andersen *et al.* [3] did not focus on the validation of their PoP maps results, rather they presented the impact of different aspects of their clustering algorithm on the results. The PoP level maps were in fact used to validate the clustering results.

The iPlane PoP level maps [31], which are mostly based on the Rocketfuel's approach, focus their validation efforts on the inter-PoP connectivity. The validation uses measurements taken from 37 Planet Lab nodes to destinations in 100 random prefix groups. The first step in the validation is end-to-end latency error estimation. Next, the two path based and latency based delay estimations are compared to the results of Vivaldi [6]. They find that 73% of their predictions obtained using the path composition approach are within 20 ms of the actual latency.

DIMES [12] validation efforts are mainly divided between two aspects of the PoP maps: the PoP extraction and its geolocation. On the extraction level, the stability of the algorithm is evaluated, as well as the best time period for maps generation. Two weeks period is found to be both with a high level of network coverage and yet flexible enough to reflect changes in the network. As the extraction is measurement based, the effect of repeatedly targeting specific networks and adding iPlane's PoP IP addresses to the measurements target was evaluated and shown to improve PoPs detection in small networks, but have a very small effect in large ASes. The assigned location of the PoPs was confirmed by several commercial ISPs, one of them a large global provider. Further comparison of maps was manually conducted with maps published on major ISPs websites, such as Sprint, QWest, AT&T, British Telecom and more. The location of research facilities' PoPs, which is known, was also validation for 50 such institutes. 49 out of the 50 were correctly located within 10km from the actual location, and the last one failed due to an error in two of the three geolocation databases used.

AS eyeballs [41] was validated by comparing the AS eyeballs results with public PoP maps information published by 45 ASes. The scope of the averaging done using the KDE method is controlled by the bandwidth of a kernel function. The validation showed that when kernel bandwidth was 40km, for 60% of ASes only 20% of the PoP locations matched the service provider's map. However, for the top 10% ASes the locations match was over 50%. On the average, 41% of

the PoP locations matched the location on the reference ISP's map. Increasing the kernel bandwidth to 80km increased the match to 60%, but decreased the number of PoPs found. Rasti *et al.* found that two causes for inaccuracy in their approach were the existence of multiple PoPs within a short distance and the placement of some PoPs away from major end users concentrations. They also compared their map with DIMES' map and found that for 80% of the eyeball ASes, the identified PoPs were a superset of DIMES'.

The Internet Topology Zoo [28] maps originate at the network operators, and are thus considered reliable. While an ISP may present a somewhat simplified network map, this aspect can be considered negligible. A possible concern is the accuracy of maps' translation into transcripts: The maps are manually annotated by the project's team, with one researcher doing the annotation and another reviewing his work, however both works are manual. The project also omits large networks with graphic links that are tangled or hard to follow.

For all the cases presented above, the validation of the generated PoPs was a very hard task: While service providers provide graphic maps of their PoPs, the PoP's actual details and the address range used within the PoP's routers are being kept confidential. PoP maps are therefore best validated when checked by the ISP, yet this is not possible on a large scale map.

6 Discussion

As we have shown in Section 2, extracting PoP level maps was originally considered as a simple task. As years passed, our understanding of the difficulties in PoP extraction grew. The approach of PoP classification using DNS, used by the Rocketfuel project and Andersen *et al.*, was shown to fail as times goes by, and less routers respond to name queries or alternatively contain inaccurate information [56]. In addition, incorrect DNS resolution leads to discovery of inter-PoP links that do not actually exist [50]. Furthermore, these works were limited in span and covered only a small portion of the network.

Of all the works presented before, only iPlane and DIMES generate PoP level maps on a periodic basis and on a large scale: iPlane update their map on a bi-monthly basis, while DIMES generate bi-weekly maps. Both mapping efforts attempt to cover the entire Internet and not only a specific ISP or a region. The two works present two different extremes of the accuracy-coverage trade-off: iPlane tries to cluster as many routers as possible into PoPs, and may include some non-PoP IPs, whereas DIMES extracts less PoPs but with a very high level of certainty that a discovered PoP includes only IP addresses belonging to PoPs. Singleton (IP addresses with a single low delay link to a PoP) which used to be part of the DIMES PoP level maps, are omitted in their newer maps to avoid assigning end-user IPs to PoPs. Consequently, the iPlane maps are considerably larger than DIMES ones, but the accuracy of mapping IP to PoP is lower. Each map can therefore be used for different types of analysis, depending on the research question.

PoP level maps can be used for a variety of applications. Understanding network topology and dynamics is one clear usage, as was done by Spring *et al.* [49]. Teixeira *et al.* [50] used PoP level topologies to study path diversity in ISP networks. The PoP level maps can also be used to evaluate and validate results of other properties of the networks, as done by Andersen *et al.* [3] who used them to check their clustering algorithm. Several works have considered the PoP level topology for delay estimation and path prediction [30,31].

A new look at the Internet's topology is through dual AS/PoP maps: maps of the Internet that combine both the AS and the PoP level graph views, leveraging the advantages of each level of aggregation. One application of dual AS/PoP maps is the study of types of relationships between ASes. Using the geographical location of PoPs, one can explore not only the connectivity between ASes on the PoP level, but also how the relations between service providers change based on the location of the PoPs. Some work in this field was done by Rasti *et al.* [41], who looked at AS connectivity at the "Edge" in AS1267 (Infostrada) and AS8234 (RAI). They found that actual peering is significantly more complex than expected, e.g., a single PoP may use five peering PoPs in different ASes for upstream. Another application is distance estimation: instead of using router-level path stitching, one can find the shortest path between every two nodes on the dual map. The shortest path can then be used to find the distance between the two nodes. PoP level maps reduce the number of edges used for the path stitching, as multiple routers are aggregated into a single PoP, and the delay-based distance estimation is more accurate as the delay estimation of a PoP level link is better than that of a single IP-level edge. Last, the PoP location can be used to improve geolocation of each node and thus the distance estimation between the pair of nodes.

PoP level maps may also be useful for research related to homeland security. Schneider *et al.* [43] used DIMES' PoP level maps to study the mitigation of malicious attacks on networks. They considered attacks on Internet infrastructure and found that cutting the power to 12% of the PoPs and 10% of power stations will affect 90% of the networks integrity. Following, they suggested ways to improve the robustness of the network by using link changes.

Annotating the PoP level maps with geographic, economic and demographic information, one can achieve an understanding of the dynamics of the Internet's structure at short and medium time scales, in order to identify the constitutive laws of Internet evolution. These can be used to develop a realistic topology generator and a reliable forecast framework that can be used to predict the size and growth of the Internet as economies grow, demographics change, and as-yet unattached parts of the world connect.

7 Conclusion

In this chapter we presented Internet PoP level maps and surveyed related works. While PoP level maps provide a good view of the network, annotated with geographic location, only a few works focused on the generation of such maps,

and currently only two projects provide large scale PoP level maps on periodic basis. As it is hard to corroborate the generated maps, we presented different approaches that are taken: some prefer extending the size of the map with the possibility of including non PoP IP addresses while others prefer smaller maps with a higher level of accuracy. The geographic location of a PoP is taken from geolocation databases or using measurement based tools. An error in either one can significantly affect the location annotation, thus different approaches are taken to mitigate this effect. We discussed the connectivity of generated PoP level maps and some of their characteristics. The PoP level maps have a high level of connectivity and the effect of delay measurements' errors is mitigated by the aggregation of IP level edges to PoP level links. PoP level maps have many applications in a vast range of research areas, and can be leveraged to study unexplored aspects of the network as well as its evolution.

Acknowledgments. We would like to thank Lior Neudorfer for contributing Figure 1. This work was partly supported by the Kabarnit Cyber Consortium (2012-2014) under Magnet program, funded by the chief scientist in the Israeli ministry of Industry, Trade and Labor.

References

1. Sarangworld project, <http://www.sarangworld.com/TRACEROUTE/>
2. Quova (2010), <http://www.quova.com>
3. Andersen, D.G., Feamster, N., Bauer, S., Balakrishnan, H.: Topology inference from BGP routing dynamics. In: Internet Measurement Workshop, pp. 243–248 (2002)
4. Bender, A., Sherwood, R., Spring, N.: Fixing ally's growing pains with velocity modeling. In: Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement (IMC 2008), pp. 337–342 (2008)
5. CenturyLink Business. CenturyLink network maps, <http://www.centurylink-business.com/demos/network-maps.html> (accessed: October 10, 2012)
6. Dabek, F., Cox, R., Kaashoek, F., Morris, R.: Vivaldi: a decentralized network coordinate system. In: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM 2004, pp. 15–26 (2004)
7. Digital Envoy. NetAcuity Edge (2010), http://www.digital-element.com/our_technology/edge.html
8. DIMES. Distributed Internet Measurements and Simulations, <http://www.netdimes.org/>
9. Donnet, B., Friedman, T.: Internet topology discovery: a survey. IEEE Communications Surveys Tutorials 9(4), 56–69 (2007)
10. Doyle, J.C., Alderson, D.L., Li, L., Low, S., Roughan, M., Shalunov, S., Tanaka, R., Willinger, W.: The robust yet fragile nature of the internet. Proceedings of the National Academy of Sciences of the United States of America 102(41), 14497–14502 (2005)

11. Eriksson, B., Barford, P., Sommers, J., Nowak, R.: A Learning-Based Approach for IP Geolocation. In: Krishnamurthy, A., Plattner, B. (eds.) PAM 2010. LNCS, vol. 6032, pp. 171–180. Springer, Heidelberg (2010)
12. Feldman, D., Shavitt, Y., Zilberman, N.: A structural approach for PoP geolocation. *Computer Networks* 56(3), 1029–1040 (2012)
13. Geobytes. GeoNetMap (2010), <http://www.geobytes.com/>
14. Georgatos, F., Gruber, F., Karrenberg, D., Santcroos, M., Uijterwaal, H., Wilhelm, R.: Providing active measurements as a regular service for ISPs. In: Proceedings of the Passive and Active Measurements Workshop, PAM 2001 (2001)
15. Govindan, R., Tangmunarunkit, H.: Heuristics for internet map discovery. In: Proceedings of IEEE INFOCOM, Tel Aviv, Israel (2000)
16. Greene, B.R., Smith, P.: Cisco ISP Essentials. Cisco Press (2002)
17. Gueye, B., Uhlig, S., Fdida, S.: Investigating the Imprecision of IP Block-Based Geolocation. In: Uhlig, S., Papagiannaki, K., Bonaventure, O. (eds.) PAM 2007. LNCS, vol. 4427, pp. 237–240. Springer, Heidelberg (2007)
18. Gueye, B., Uhlig, S., Ziviani, A., Fdida, S.: Leveraging Buffering Delay Estimation for Geolocation of Internet Hosts. In: Boavida, F., Plagemann, T., Stiller, B., Westphal, C., Monteiro, E. (eds.) NETWORKING 2006. LNCS, vol. 3976, pp. 319–330. Springer, Heidelberg (2006)
19. Gueye, B., Ziviani, A., Crovella, M., Fdida, S.: Constraint-based geolocation of Internet hosts. *IEEE/ACM Transactions on Networking* 14(6), 1219–1232 (2006)
20. Hexsoft Development. IP2Location (2010), <http://www.ip2location.com>
21. hostip.info, hostip.info (2010), <http://www.hostip.info>
22. Huffaker, B., Dhamdhere, A., Fomenkov, M., Claffy, K.: Toward Topology Dualism: Improving the Accuracy of AS Annotations for Routers. In: Krishnamurthy, A., Plattner, B. (eds.) PAM 2010. LNCS, vol. 6032, pp. 101–110. Springer, Heidelberg (2010)
23. Huffaker, B., Plummer, D., Moore, D., Claffy, K.: Topology discovery by active probing. In: Symposium on Applications and the Internet (SAINT), pp. 90–96. SAINT (2002)
24. Hyun, Y.: Archipelago measurement infrastructure, <http://www.caida.org/projects/ark/>
25. Internet2, Internet2 gigapop list, <http://eng.internet2.edu/gigapoplist.html> (accessed: October 10, 2012)
26. Katz-Bassett, E., John, J.P., Krishnamurthy, A., Wetherall, D., Anderson, T., Chawathe, Y.: Towards IP geolocation using delay and topology measurements. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC 2006), pp. 71–84 (2006)
27. Keys, K., Hyun, Y., Luckie, M., Claffy, K.: Internet-Scale IPv4 Alias Resolution with MIDAR. *IEEE/ACM Transactions on Networking* (2012)
28. Knight, S., Nguyen, H., Falkner, N., Bowden, R., Roughan, M.: The internet topology zoo. *IEEE Journal on Selected Areas in Communications* 29(9), 1765–1775 (2011)
29. Laki, S., Mátray, P., Hága, P., Sebök, T., Csabai, I., Vattay, G.: Spotter: A model based active geolocation service. In: Proceedings of IEEE INFOCOM, Shanghai, China (2011)
30. Lee, D., Jang, K., Lee, C., Iannaccone, G., Moon, S.: Scalable and systematic internet-wide path and delay estimation from existing measurements. *Computer Networks* 55(3), 838–855 (2011)
31. Madhyastha, H.V.: An information plane for internet applications. Thesis, University of Washington (2008)

32. Madhyastha, H.V., Anderson, T., Krishnamurthy, A., Spring, N., Venkataramani, A.: A structural approach to latency prediction. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC 2006), pp. 99–104 (2006)
33. MaxMind LLC. GeoIP (2010), <http://www.maxmind.com>
34. Merit Network. Internet routing registries, <http://www.irr.net/>
35. Muir, J.A., Oorschot, P.C.V.: Internet geolocation: Evasion and counterevasion. *ACM Computing Surveys* 42(1), 1–23 (2009)
36. Padmanabhan, V.N., Subramanian, L.: An investigation of geographic mapping techniques for Internet hosts. In: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2001), pp. 173–185 (2001)
37. Pastor-Satorras, R., Vespignani, A.: *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press (2004)
38. Poese, I., Uhlig, S., Kaafar, M.A., Donnet, B., Gueye, B.: IP geolocation databases: unreliable? *ACM SIGCOMM Computer Communication Review* 41(2), 53–56 (2011)
39. QWest Business. Network maps, <http://shop.centurylink.com/largebusiness/enterprisesolutions/networkMaps/> (accessed: October 10, 2012)
40. Radoslavov, P., Tangmunarunkit, H., Yu, H., Govindan, R., Shenker, S., Estrin, D.: On characterizing network topologies and analyzing their impact on protocol design. Technical report, Computer Science Department, University of Southern California (2000)
41. Rasti, A.H., Magharei, N., Rejaie, R., Willinger, W.: Eyeball ASes: from geography to connectivity. In: Proceedings of the 10th Annual Conference on Internet Measurement (IMC 2010), pp. 192–198 (2010)
42. Sardella, A.: Building next-gen points of presence, cost-effective PoP consolidation with juniper routers. White paper, Juniper Networks (2006)
43. Schneider, C.M., Moreira, A.A., Andrade, J.S., Havlin, S., Herrmann, H.J.: Mitigation of malicious attacks on networks. *Proceedings of the National Academy of Sciences* 108(10), 3838–3841 (2011)
44. Schwartz, Y., Shavitt, Y., Weinsberg, U.: A Measurement Study of the Origins of End-to-End Delay Variations. In: Krishnamurthy, A., Plattner, B. (eds.) *PAM 2010*. LNCS, vol. 6032, pp. 21–30. Springer, Heidelberg (2010)
45. Shavitt, Y., Zilberman, N.: A geolocation databases study. *IEEE Journal on Selected Areas in Communications* 29(9), 2044–2056 (2011)
46. Shavitt, Y., Zilberman, N.: Geographical Internet PoP Level Maps. In: Pescapè, A., Salgarelli, L., Dimitropoulos, X. (eds.) *TMA 2012*. LNCS, vol. 7189, pp. 121–124. Springer, Heidelberg (2012)
47. Siamwalla, R., Sharma, R., Keshav, S.: *Discovering Internet Topology*. Technical report, Cornell University (1998)
48. Siwbersad, S.S., Gueye, B., Uhlig, S.: Assessing the Geographic Resolution of Exhaustive Tabulation for Geolocating Internet Hosts. In: Claypool, M., Uhlig, S. (eds.) *PAM 2008*. LNCS, vol. 4979, pp. 11–20. Springer, Heidelberg (2008)
49. Spring, N.T., Mahajan, R., Wetherall, D., Anderson, T.E.: Measuring ISP topologies with Rocketfuel. *IEEE/ACM Transactions on Networking* 12(1), 2–16 (2004)
50. Teixeira, R., Marzullo, K., Savage, S., Voelker, G.M.: In search of path diversity in ISP networks. In: Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement (IMC 2003), pp. 313–318 (2003)
51. TeliaSonera International Carrier. Network map, <http://www.teliasoneraic.com/NetworkFlash/index.htm> (accessed: October 10, 2012)

52. University of Oregon Advanced Network Technology Center. Route views project, <http://www.routeviews.org/>
53. Wong, B., Stoyanov, I., Sirer, E.G.: Octant: A comprehensive framework for the geolocalization of Internet hosts. In: Proceedings of the 4th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2007 (2007)
54. XO. Complete network assets, http://www.xo.com/SiteCollectionImages/about-xo/xo-network/maps/map_complete_1600.gif (accessed: October 10, 2012)
55. Yoshida, K., Kikuchi, Y., Yamamoto, M., Fujii, Y., Nagami, K., Nakagawa, I., Esaki, H.: Inferring POP-Level ISP Topology through End-to-End Delay Measurement. In: Moon, S.B., Teixeira, R., Uhlig, S. (eds.) PAM 2009. LNCS, vol. 5448, pp. 35–44. Springer, Heidelberg (2009)
56. Zhang, M., Ruan, Y., Pai, V., Rexford, J.: How DNS misnaming distorts Internet topology mapping. In: Proceedings of the Annual Conference on USENIX 2006 Annual Technical Conference (ATEC 2006), pp. 34–34 (2006)
57. Zupan, J.: Clustering of large data sets. Chemometrics Research Studies Series. Research Studies Press (1982)

Analysis of Packet Transmission Processes in Peer-to-Peer Networks by Statistical Inference Methods

Natalia M. Markovich¹ and Udo R. Krieger^{2,*}

¹ Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
markovic@ipu.rssi.ru

² WIAI, Otto-Friedrich-University, Bamberg, Germany
udo.krieger@ieee.org
<http://www.uni-bamberg.de/ktr>

Abstract. Applying advanced statistical techniques, we characterize the peculiarities of a locally observed peer population in a popular P2P overlay network. The latter is derived from a mesh-pull architecture. Using flow data collected at a single peer, we show how Pareto and Generalized Pareto models can be applied to classify the local behavior of the population feeding a peer. Our approach is illustrated both by file sharing data of a P2P session generated by a mobile BitTorrent client in a WiMAX testbed and by video data streamed to a stationary client in a SopCast session. These techniques can help us to cope with an efficient adaptation of P2P dissemination protocols to mobile environments.

Keywords: heavy hitter model, Generalized Pareto distribution, peer-to-peer network, change-point detection.

1 Introduction

In recent years modern dissemination platforms employing peer-to-peer (P2P) overlay protocols have gained increasing interest. They have been derived from BitTorrent and its ramifications, for instance, GoalBit, Zattoo, PPLive, SopCast, Vodddler or Skype (c.f. [4], [22], [24]). The latter have become mature middle-ware systems to distribute real-time media streams among interested clients. The estimation of the traffic matrices arising from such P2P based streaming or file sharing services and the integration of these services into mobile environments demand a solid analysis of the generated packet streams and the required capacity on an IP underlay network (cf. [15], [27]).

Components of P2P teletraffic engineering comprise basic elements such as monitoring and analysis of overlay and underlay structures, the design of topology aware routing and QoS driven peer- as well as piece-selection algorithms and the modeling of the resulting workloads (cf. [3], [7], [13], [17], [25], [26]). The

* The authors acknowledge the partial financial support by the ESF-project COST IC0703.

latter tasks involve the statistical characterization of P2P packet flows and the determination of an effective bandwidth regarding aggregated P2P traffic on IP network links. For this purpose the required delay-loss profiles of a media service such as the tolerable packet and frame losses and the playback delay bounds of the media players should be taken into account.

Compared to a parametric teletraffic approach, purely measurement based concepts may provide an alternative. The latter can be integrated easily into real-time control components of the P2P middleware and handle the evolution of the P2P protocols more rapidly. To respond to this fast deployment of P2P overlay structures, we have developed a comprehensive P2P traffic measurement, modeling and teletraffic analysis concept (cf. [20]). It integrates four orthogonal dimensions to cope with the analysis of P2P structures and P2P traffic characterization: (1) traffic measurements at the packet layer combining passive and active monitoring techniques by the open source tool Atheris [1], [8], (2) data extraction, analysis and inspection of P2P overlays based on a hierarchical multi-layer modeling concept [20], (3) a characterization of the overlay structure by techniques and metrics of complex networks, and, last but not least, (4) a non-parametric teletraffic modeling approach based on the statistical characterization of P2P traffic [19]. The latter relies on the analysis and estimation of the bivariate distribution $\mathbb{P}\{X_i \leq x, Y_i \leq y\}$ of packet inter-arrival times X_i and packet lengths Y_i extracted from corresponding aggregated flows of P2P conversations and their collected i.i.d samples $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$.

Considering the control plane of a P2P system at the side of mobile clients, these building blocks constitute basic elements of a middleware concept derived from observation-driven, sound control-theoretical design principles. The latter should automatically adapt itself to the states of a very dynamic mobile environment, in particular to handover management, using the fundamental feedback principles of a Luenberger state observer and modeling results of teletraffic theory (cf. [3], [7], [9], [23]). To limit complexity, the information of the controls should only stem from local observations at the packet and session layers of the clients. Our recently developed system RapidStream [2] which offers a P2P video service for Android smartphones illustrates this necessity (cf. [10]). By these means we want to enhance the heuristic control approaches developed in the area of P2P protocols. Improvements concerning adaptive peer-selection and chunk scheduling can be implemented into an enhanced version of the prototype Rapidstream and its performance can be evaluated by the distributed monitoring capabilities of a currently developed enhanced variant of Atheris [1].

It is the objective of this paper to elaborate on new building blocks of our statistical approach. It is motivated by a statistical characterization of the local peculiarities of a peer population feeding a single client. The latter concept can be combined with the bandwidth measurement functionality of the Atheris approach [8] and a control-theoretical concept to identify dynamically the most important peers and to adapt peer-selection policies in mobile networks. These peers should provide a sufficiently large packet input rate to feed the media player

without performance degradation (cf. [4], [9]). This limitation to useful peers is of particular importance if a mobile environment is considered (cf. [10], [11]).

We have partially validated the developed classification concept by experimental testbeds collecting a rich set of traces (see, e.g., [9], [15], [20]). Due to the renewal process character of chunk request-reply patterns during the lifetime of a P2P session we have seen that one does not gain more information if a huge set of traces is collected. The basic locally observed dissemination features of a P2P protocol that we want to investigate are already manifested in small sets. Here we use two traces as an example to illustrate some of these results. On one hand, a trace of P2P streaming traffic observed at a monitored stationary SopCast client has been used as representative of a high-speed wireline network (cf. [20]). On the other hand, a P2P file sharing trace collected in a WiMAX testbed in Seoul, Korea, is evaluated as example of a wireless environment (cf. [9], [15]). There a mobile BitTorrent client has been traveling in a bus. Regarding the provisioning of mobile high-speed services in an urban environment, this scenario provides an interesting example. This Asian setting is different from the normal European one and WiMAX an interesting competitor of UMTS or LTE networks. Based on the corresponding measurement studies we provide further characteristics of the latter peer-to-peer file sharing and streaming services. The data analysis illustrates the features of our analysis approach, partly validates former findings (cf. [9], [26]) and extends our concepts on P2P traffic characterization and measurement-driven classification of a locally observed peer population by adequate teletraffic models.

The rest of the paper is organized as follows. In section 2 the statistical analysis and modeling of packet flows of a locally observed peer population in terms of a Pareto model is presented. It is illustrated by superimposed packet flows feeding a monitored SopCast client in an observed P2P session. In section 3 the characterization of the packet transmission processes which are locally observed at a mobile client of a typical BitTorrent session in a WiMAX environment by means of a generalized Pareto model is stated. Principal component analysis is applied to prove that both volume and count based attributes of the monitored flows will provide an equivalent classification of the locally activated peer population. Finally, some conclusions on peer-to-peer protocols and their application to mobile networks are drawn.

2 Characterizing the Size-Count Disparity of a P2P Session by a Pareto Model

In the following we consider overlay networks derived from the mesh-pull architecture of a modern P2P dissemination service and use streaming data of a SopCast session as illustrative example (cf. [20]). Such an overlay network normally embeds a monitored peer, here called home peer p_0 , requesting a certain object like a media file or a video stream into a dense mesh of n feeding peers p_i from a finite peer population \mathcal{U} which share common interest in a specific object.

In a streaming context this strategy will guarantee a reliable and timely supply of chunk sequences to the streaming engine of a host. For instance, our

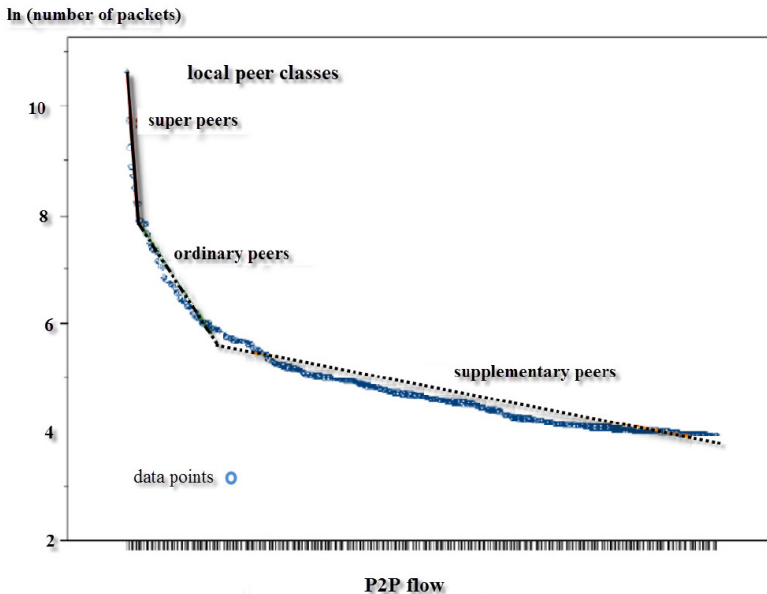


Fig. 1. Classification of peer relations based on the packet flows of all active peers feeding a stationary home peer during a SopCast session (cf. also [20, Fig. 8])

analysis [20] that will be used as illustrative example and subsequently be further extended has revealed that SopCast clients are typically connected to up to thousand different peers during the lifetime of a session and the exchanged flow data exhibit a hierarchical pattern.

This feature and the instantiated packet flow relations have been observed for other P2P applications as well and are caused by the applied peer and piece selection strategies of the underlying P2P protocol (cf. [7], [26]). Therefore, it is necessary to investigate the preference relationship among the packet flows of the active peers in order to understand the inherent local hierarchical structure of the mesh-pull topology. As basic metric one can use the number of transferred packets of all active flows feeding the home peer p_0 and the intensity or the volume of these flows which are depicted on a logarithmic scale.

Our previous investigations have revealed that the first criterion, for example, is simple to monitor and allows us to distinguish three local levels of peers $p_i \in \mathcal{U}$ associated with a home peer $p_0 \in \mathcal{U}$ during a session, namely his individual *super peers*, *ordinary peers* and *supplementary peers* (see fig. 1, also [20, Fig. 8]). Such hierarchical flow patterns are observed for other protocols as well, and thus bear a certain generality.

The investigation of the transferred accumulated volumes among these n active peers of the realized flow graph G_V , in particular the number of packets and the byte volumes flowing inbound and outbound to a monitored home peer p_0 , obviously reveals the local hierarchical structure of the overlay and expresses the implemented selection strategies (see fig. 1, cf. [20], [26]).

We intend to describe these observed structures by a universal teletraffic model that can be used in further design studies or become a load model of other performance investigations, e.g. in a simulation of mobile clients. If we arrange the n flows according to the number of exchanged packets on a logarithmic scale, we can realize the hierarchy of the locally instantiated peer classes. Interpreting the relative number of transferred packets as frequency f_i to select a feeder p_i , we can model the ranked selection process by a versatile heavy-tailed distribution of a random variable (rv) Y on the integers \mathbb{N} . It should obey a distribution function (df) of a generalized Zipf type. We may choose, for instance, a special case of the zero-truncated Lerch distribution with probability mass function (pmf)

$$p_k = \mathbb{P}\{Y = k\} = C \cdot \frac{p^k}{(a+k)^\alpha}, \quad k \in \mathbb{N}, \quad (1)$$

the parameters $a > -1$, $p \in (0, 1]$, and the tail coefficient $\alpha \in \mathbb{R}$. The normalization constant $C = [p\Phi(p, a+1, \alpha)]^{-1}$ is defined in terms of the Lerch' transcendent $\Phi(p, a, \alpha) = \sum_{k=0}^{\infty} p^k \cdot (a+k)^{-\alpha}$, (cf. [5], [28]). Selecting the parametrization $p = 1, a \geq 0, \alpha > 1$, we get the Zipf-Mandelbrot pmf $\mathbb{P}\{Y = k\} = C \cdot (a+k)^{-\alpha}$, $k \in \mathbb{N}$, $C^{-1} = \Phi(1, a+1, \alpha)$ and with the restriction $a = 0$ the well-known Zipf law

$$\mathbb{P}\{Y = k\} = C \cdot \frac{1}{k^\alpha}, \quad k \in \mathbb{N}, \quad C^{-1} = \Phi(1, 1, \alpha). \quad (2)$$

If we plot the rank-frequency relationship of the relative number f_i of packets of all inbound flows $\phi(p_i, p_0)$ transferred from a peer p_i to the home peer p_0 on a log-log scale, we can identify in this case a linear relation between $\ln f_i$ and $\ln i$ of the related ranks i of the feeders p_i (see also [20]). Thus, a pmf of Zipf type (2) will adequately describe the local hierarchical peer structure seen by the home peer.

In our example of a SopCast session depicted in fig. 1 this feature has been validated (cf. [20]). Subsequently, we will see that the basic BitTorrent protocol itself, which has inspired the development of SopCast, behaves slightly different and must be modelled in a modified manner.

If we interpret the transferred number of packets of a flow $\phi(p_i, p_0)$ as realization x_i of an equivalent income $X_i \in \mathbb{R}$ of the feeding peer $p_i, i \in \{1, \dots, n\}$, we can represent the local P2P packet model of a session by a corresponding heavy-tailed physical model of Pareto type with a rv X and its sample $\{X_1, \dots, X_n\}$ (cf. [20], [21]). We denote the distribution function of this Pareto model associated with the Lerch ranking law by

$$F(x) = \mathbb{P}\{X \leq x\} = \int_{x_0}^x f(t)dt = 1 - Cx^{-\beta} = 1 - \left(\frac{x}{x_0}\right)^{-\beta} \quad (3)$$

with $x \geq x_0 > 0$ and tail index $\beta = 1/\alpha$.

The corresponding $q = 1 - p$ quantile function of this Pareto model $F_q(x_p) = 1 - F(x_p) = p \in (0, 1)$, i.e. $x_p = F_q^{-1}(x_0, p, \beta) = x_0 \cdot p^{-1/\beta}$, can be used to define the local classes of the feeding peers. Taking, for instance, the 2.5% or 5% as well as 10% and 20% levels for p , their quantiles $x_{0.025}, x_{0.5}, x_{0.1}, x_{0.2}$ may specify the break points of the local classes of super peers, ordinary and supplementary peers associated with a home peer. Alternatively, these break points may be determined automatically by change-point detection algorithms (see [12]).

Using the transferred numbers of packets $x_1 \geq x_2 \dots \geq x_n$ of the flows or approximating the pmf $p_i = \mathbb{P}\{Y = i\}$ by the empirical values $\{f_i, i = 1, \dots, n\}$ of the n flows, we can estimate the tail index β by Hill's estimate $\hat{\beta}^{-1} = \frac{1}{n-1} \left(\sum_{k=1}^{n-1} \ln\left(\frac{x_k}{x_n}\right) \right) = \frac{1}{n-1} \sum_{k=1}^{n-1} \ln(x_k) - \ln(x_n)$ or in terms of Newman's estimate $\hat{\alpha} = \frac{1}{n} \sum_{k=1}^n \ln\left(\frac{f_i}{f_{\min}}\right)$ where $f_{\min} = f_n$ represents the minimal measured value (cf. [18], [21]). It corresponds to the value of the gradient of the linear segment in the rank-frequency plot (cf. [20]).

To investigate the size-count disparity issues in more detail, we can apply the excess wealth function $W_X(\cdot)$ (cf. [16]). Regarding the Pareto model (3) or any heavy-tailed, nonnegative income variable X with finite first moment $\mu = \mathbb{E}(X)$ it is determined by the corresponding continuous distribution function $F(x)$ in terms of the excess wealth transform

$$W_X(q) := \int_{x_q}^{\infty} (1 - F(x)) dx = W_X(F(x_q)) \tag{4}$$

of X . Here $x_q = F^{-1}(q), q \in (0, 1)$ is the q th quantile of $F(x)$. The random variable $X_{ew} := W_X(F(X)) = \int_X^{\infty} (1 - F(t)) dt$ defined for $F(X) = q \in (0, 1)$, which is a uniformly distributed rv U on $(0, 1)$, is called observed excess wealth. By the underlying integral transform of the survival function $1 - F(x) = \mathbb{P}\{X > x\}$ of X ,

$$\pi_X(t) := \int_t^{\infty} (1 - F(x)) dx = W_X(F(t)), \quad t \geq 0, \tag{5}$$

called stop-loss transform, it determines the observed excess wealth $X_{ew} = \pi_X(X)$. Hence, the survival function of X_{ew} can be simulated easily since it is the inverse of the excess wealth transform, i.e.

$$\mathbb{P}\{X_{ew} > y\} = \mathbb{P}\{W_X(U) > y\} = \mathbb{P}\{U \leq W_X^{-1}(y)\} = W_X^{-1}(y), \quad y \in (0, \mu).$$

Then we can conclude that for the Pareto model (3), or any other heavy-tailed model, the normalization by the mean is determined by

$$W_N(x_p) = \frac{W_X(1-p)}{\mu} = \frac{\pi_X(x_p)}{\pi_X(x_0)} = \frac{\int_{x_p}^{\infty} x f(x) dx}{\int_{x_0}^{\infty} x f(x) dx} = \left(\frac{x_p}{x_0}\right)^{-(\beta-1)}. \tag{6}$$

Here x_p denotes the $q = 1 - p$ quantile, i.e. $\mathbb{P}\{X > x_p\} = p \in (0, 1)$. It yields a size weighting of the wealth excess beyond x_p . It can be easily used to study size-count disparity issues in more detail by simply fitting the tail index β in (3) to the available data (cf. [20]).

In our context it means that the fraction of those packets sent by the most active part of the peers p_i is specified in terms of (6) by the ratio of the packet load of the fraction of flows exceeding level x_p compared to the total packet load. Hence, given the Pareto model (3) the most active $p \cdot 100$ % of the flows determine the fraction $W_N(x_p)$ of the sent packets by means of the $1 - p$ th quantile x_p in terms of $W_N(x_p) = p^{(\beta-1)/\beta}$. The latter term is also related to the Lorenz curve (cf. [21]).

Motivated by this analysis, we subsequently investigate as an alternative a parametric analysis and classification method based on a generalized Pareto model. We shall illustrate its relevance by means of the fundamental BitTorrent protocol in a mobile network environment.

3 Modeling Local Peculiarities of a Peer Population by a Generalized Pareto Distribution

In the previous section we have shown that the traffic flows which are realized in a P2P overlay network by a population $\mathcal{U}_n := \{p_0, p_1, \dots, p_n\} \subseteq \mathcal{U}$ of active clients generate a characteristic load pattern at the single measurement point of an involved, monitored client p_0 during a P2P session. The classification of the observed population of feeding peers from $\mathcal{U}_{p_0} := \mathcal{U}_n \setminus \{p_0\}$ can be based on this information and determine an understanding of the underlying flow graph G_V of the P2P network (see [20], [26]). Since we can record different statistical attributes of the flows exchanged with the monitored client, such as the number of incoming or outgoing packets, the overall transferred packets, or the exchanged byte volumes, the question arises whether there is a most informative conversation attribute which is arising from the collected flow statistics.

3.1 Principal Component Analysis of the Exchanged Flow Data

Let R_i denote the number of packets that a monitored home peer p_0 has received from a feeding peer $p_i \in \mathcal{U}_{p_0}$ in the overlay network during a typical P2P session. Let $V_i^{(i)}$ be the volume of the inbound traffic received from peer p_i , $V_i^{(o)}$ be the volume of the outbound traffic sent from p_0 to peer p_i and $V_i^{(e)}$ be the volume of the overall traffic exchanged with peer p_i . We assume that the random variables of each attribute are governed by a common distribution and denote the underlying generic random variables by $R, V^{(i)}, V^{(o)}, V^{(e)}$, respectively. The classification of the observed population of feeding peers $p_i \in \mathcal{U}_{p_0}$ can be based on this information.

To investigate the issue whether there is a most informative entity in the flow statistics, one should study the potential correlation among the attribute variables of the packet flows feeding a home peer p_0 during a P2P session. If the volume oriented random variables $Y \in \{V_i^{(i)}, V_i^{(o)}, V_i^{(e)}\}$ and the simple counting statistics R_i are linearly correlated, then there should exist a corresponding linear function $y = f_k(x) = a_k x + b_k$, $k \in \{i, o, e\}$ such that the regression

$$V^{(k)} = f_k(R) = a_k R + b_k, \quad k \in \{i, o, e\} \quad (7)$$

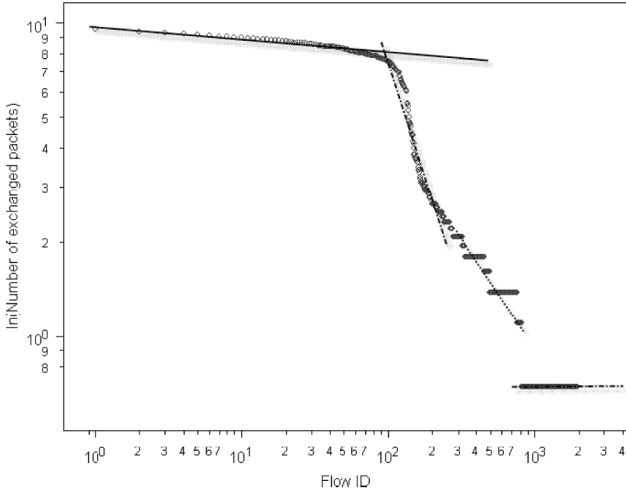


Fig. 2. Exchanged flow data of the WiMAX bus trace on a double logarithmic scale

holds. In this case a visual representation of the relationships $(R, V^{(k)})$, $k \in \{i, o, e\}$ should arrange the data of the peer flows along straight lines.

Principal component analysis (PCA) or other methods arising from factor analysis can be applied to study this issue in a rigorous manner.

3.2 Application to BitTorrent Data of a WiMAX Network

In the following we use the flow data captured by some BitTorrent bus trace in the WiMAX testbed in Seoul on March 17, 2010, as illustrative example (cf. [9], [15]). Studying the packet flows ϕ_i exchanged among the home peer p_0 and a feeder p_i in fig. 2, we see that there are at least four different tail regimes if we depict the ordered flow data on a double logarithmic scale. The shown structure illustrates that only the top 500 flows of the first two classes of super peers and dominant peers provide relevant information on the classification of the most dominant feeders within the peer population. Therefore, we will focus on the latter subset of the peer population in the subsequent investigations.

Applying the regression concept (7) to all flows ϕ_i of packets exchanged with the home peer p_0 that are arising from a feeder p_i and captured by the BitTorrent bus trace, for instance, we have realized (see [12, Fig. 3]) that there exists a perfect linear matching among all these variables as expected.

A factor analytic study of this data set using principal component analysis shows us that 99.49% of the variance in the flow data is explained by one major principal component depending on the four attributes $R, V^{(k)}$, $k \in \{i, o, e\}$. Each attribute of the flow data contributes between 24.7 to 25.1% to this dimension of the dominant principal component 1. The other components simply reflect the influence of the volume variables $V^{(k)}$, $k \in \{i, o, e\}$. Since all attributes nearly

equivalently explain the variance of the dominant principal component, we see that there is no preferred variable carrying more information than all others. It confirms the result of the dependence plot in [12, Fig. 3].

In conclusion, we see that the simplest attribute ' R_i =number of packets incoming from (or exchanged with) a certain peer $p_i \in \mathcal{U}_{p_0}$ ' is sufficient to perform a sound classification analysis. Hence, a measurement campaign aiming to achieve the classification objective can be based on this simple packet counting metric R . There is no loss of information to apply it as foundation of a local peer classification approach and for further control actions in a mobile setting.

3.3 Generalized Pareto Modeling of Data Exchange Patterns

In the following we develop a more general parametric method to classify the exchange patterns within a local peer population \mathcal{U} of an overlay network feeding a monitored home peer $p_0 \in \mathcal{U}$. We are looking for a mathematically sound procedure which can generate in a very efficient manner a reasonable partitioning of the set of all active peers $\mathcal{U}_n \subseteq \mathcal{U}$ feeding the observed home peer p_0 by means of their packet flows.

From a statistical point of view, we consider a sample $\mathcal{X}^{(n)} = \{X_1, X_2, \dots, X_n\}$ of iid random variables $X_i, i \in \{1, \dots, n\}$, which are governed by a common distribution function $F(x)$ of an underlying generic rv X . If we interpret X_i as number of transferred packets to or from peer $p_i, i \in \{1, \dots, n\}$, during a monitored session, we have seen in the previous section that there is evidence that the related df obeys a heavy-tailed law of Pareto-type, i.e. $1 - F(x) = \mathbb{P}\{X > x\} \sim l(x)x^{-1/\gamma}$, for large $x \in \mathbb{R}^+$. Here $l(x)$ is a slowly varying function satisfying $\frac{l(tx)}{l(x)} \rightarrow 1$ as $x \rightarrow \infty, \forall t > 0$, and $f(x) \sim g(x)$ denotes the asymptotically identical growth rate, $\lim_{x \rightarrow \infty} f(x)/g(x) = 1$, of f and g (cf. [18, Def. 10, p. 4]). The positive tail index $\alpha = 1/\gamma > 0$ characterizes the slow decay of the tail of the distribution at infinity as basic model parameter. Different estimation techniques can be applied to determine the corresponding extreme-value index (EVI) $\gamma > 0$ by means of the sample (cf. [18, §1.2, p. 6f]).

Following classical extreme-value theory, we consider the absolute excess $X - u$ of the heavy-tailed rv X over a high threshold $u > 0$ subject to the condition of the exceedence $X > u$, i.e. the conditional rv $Z := X - u \mid X > u$. By the peaks-over-threshold method (POT) of extreme-value theory [18, p. 14], in particular Pickands' theorem, we can conclude that for increasing thresholds $u \rightarrow \infty$ the asymptotic distribution of rv Z is determined by a Generalized Pareto df (GPD(γ, σ))

$$\Psi(z) = \mathbb{P}\{Z \leq z\} = 1 - (1 + \gamma z/\sigma)^{(-1/\gamma)}, \quad z \geq 0. \quad (8)$$

Here the scaling parameter $\sigma > 0$ is depending on the threshold u , i.e. $\sigma(u)$, and the EVI $\gamma > 0$ can be determined by the sample (cf. [6], [18, Sec. 1.2.3, p. 13]). Then we know that the transformation

$$Y = \frac{1}{\gamma} \ln \left(1 + \frac{\gamma}{\sigma} Z \right)$$

yields an exponentially distributed rv with unit scale parameter $\lambda = 1$, i.e. $\mathbb{P}\{Y \leq y\} = 1 - \exp(-y), y \geq 0$, (cf. [14, Chap. 19.5.5, p. 240]).

Let us assume that Z obeys a GPD(γ, σ) law. Then we see that

$$\begin{aligned} \mathbb{P}\{Z - w \leq z \mid Z > w\} &= \frac{\mathbb{P}\{w < Z \leq z + w\}}{\mathbb{P}\{Z > w\}} = \frac{\Psi(z + w) - \Psi(w)}{1 - \Psi(w)} \\ &= \frac{(1 + \gamma w/\sigma)^{(-1/\gamma)} - (1 + \gamma(z + w)/\sigma)^{(-1/\gamma)}}{(1 + \gamma w/\sigma)^{(-1/\gamma)}} = 1 - (1 + \gamma z/(\sigma + \gamma w))^{(-1/\gamma)} \end{aligned} \tag{9}$$

holds and, hence, for all $w > u > 0$ the conditional rv $Z(w) := Z - w \mid Z > w$ satisfies also a GPD law with the modified scale parameter

$$\sigma(w) = \sigma + \gamma w > 0 \tag{10}$$

depending in a linear manner on the EVI γ . It means that we can consistently study GPD-like tail models of our original sample for all higher thresholds $w \geq u$ after identifying an appropriate initial threshold value $u > 0$, e.g. as appropriate high quantile of the empirical distribution of the flow data, such that the GPD hypothesis approximately holds. For this purpose we have to apply a linear scaling (10) of the initial scale parameter σ (see [6], [18]).

Thus we can consider a sample $\mathcal{X}^{(n)} = \{X_1, X_2, \dots, X_n\}$ of n iid heavy-tailed rvs such as the number of packets R_i that a monitored home peer p_0 has received and that were sent by a feeding peer $p_i, i \in \{1, \dots, n\}$. Then we select for a predetermined threshold $0 < u \in \mathbb{R}$ the corresponding sequence of N_u exceedences $\{X_{i_1}, \dots, X_{i_{N_u}}\}$, where

$$\begin{aligned} i_1 &:= \min\{i \in \{1, \dots, n\} \mid X_i > u\} = \min\{i \in \{1, \dots, n\} \mid \mathbf{1}_{(u, \infty)}(X_i) = 1\} \\ i_{j+1} &:= \min\{i \in \{1, \dots, n\} \mid i > i_j, \mathbf{1}_{(u, \infty)}(X_i) = 1\}, \quad j = 1, \dots, N_u - 1 \\ N_u &:= \sum_{i=1}^n \mathbf{1}_{(u, \infty)}(X_i) \end{aligned}$$

determines the latter indices of the sample.

Based on this information we can estimate the unconditional tail distribution $\bar{F}(x) = \mathbb{P}\{X > x\}$ by the sample $\mathcal{X}^{(n)}$ in terms of $\hat{H}(x) = \frac{N_u}{n} \left(1 + \frac{\hat{\gamma}}{\hat{\sigma}}(x - u)\right)$ and the logarithmically transformed excess process $\hat{Y} = \frac{1}{\hat{\gamma}} \ln\left(\frac{n}{N_u} \hat{H}(x)\right) = \frac{1}{\hat{\gamma}} \ln\left(1 + \frac{\hat{\gamma}}{\hat{\sigma}}(x - u)\right)$ (see [12], [18, (1.18), p. 14]). Here estimates $\hat{\gamma}$ of the EVI γ and $\hat{\sigma}$ of the scale parameter σ of the GPD are used (see (8), (10)).

3.4 Application to BitTorrent Data of a WiMAX Bus Trace

We will illustrate the classification concept derived from this generalized Pareto model by flow data exchanged with a monitored home peer in the overlay network of the swarm-like P2P protocol BitTorrent. We are again regarding a client moving by bus through Seoul on March 17, 2010, and consider the data set collected in the WiMAX testbed as representative example (cf. [9], [15]).

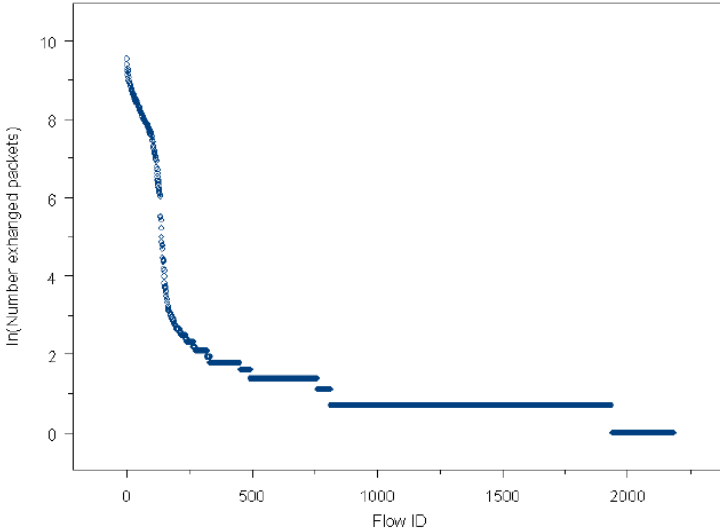
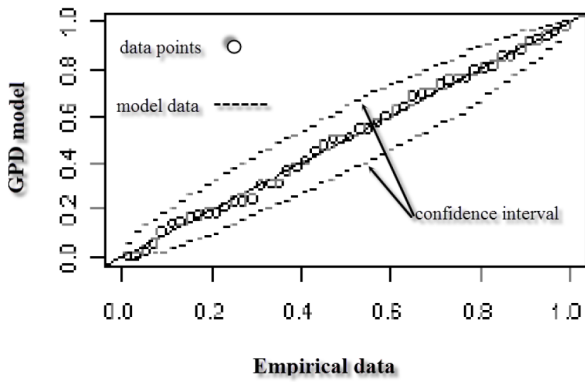


Fig. 3. Ln-transformed number of packets R_i exchanged with a peer p_i by a flow ϕ_i

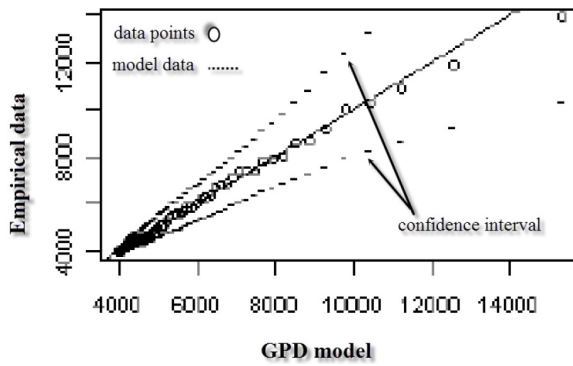
A plot of the ordered number of packets R_i exchanged with active peers $p_i \in \{1, \dots, n\}$ during this session characterized by a flow identifier i and its logarithmic transformation $\ln R_i$ in figure 3 illustrates that we can expect a heavy-tailed behavior of the most dominant peers due to the piecewise linear shape. Moreover, we see that there occurs a substantial change in the flow behavior within a part of the dominant flows (on the lhs in fig. 3). It is due to the separation of the feeding peer population into the groups of super peers, ordinary peers and the supplementary ones (see [9]).

An analysis of the dominant flows by a peaks-over-threshold (POT) methodology with a relatively high threshold u should reveal this behavior. Here we have chosen $u = 4000$ exchanged packets as separator of the flows. A fit of the available packet flow data $R_i, i = 1, \dots, n = 2184$ of the WiMAX trace by the maximum likelihood method implemented by the routine 'fitgpd' of the R-package POT yields the parameter estimates $\hat{\sigma} = 2344, \hat{\gamma} = 0.02107$ of a Generalized Pareto distribution regarding the generic conditional excess variable $Z = R - u \mid R > u$ of exchanged packets. A related visual comparison of the counted flow data and the GPD model by a probability plot and a QQ plot with associated confidence intervals in fig. 4 illustrates that the data of all dominant flows are relatively well covered by the GPD distribution (see also [12, Sec. 3.2]).

If the threshold is chosen even higher, e.g. $u = 6000$, the data can also be covered by a GPD model, as shown in fig. 5 by the plots of the conditional excess and the tail distribution $\mathbb{P}\{Z \geq x\}$, arising from the data and the fitted GPD model. However, now the parameters of the fitted model have slightly changed as indicated by (9).

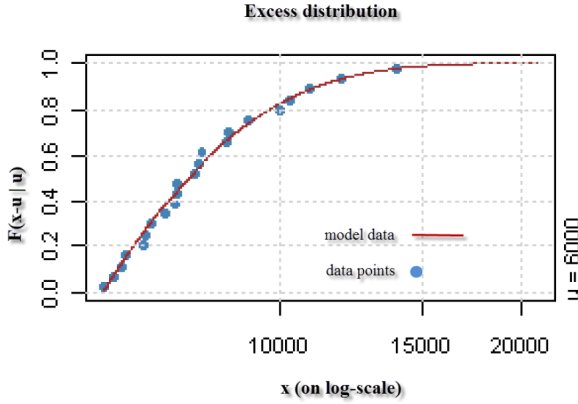


(a) Comparison by a probability plot

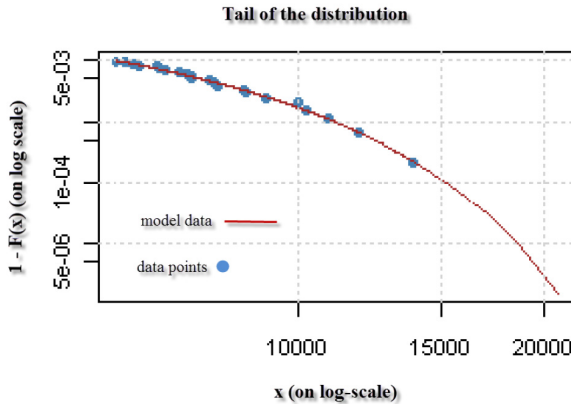


(b) Comparison by a QQ plot

Fig. 4. Fitted GPD tail model for threshold $u = 4000$



(a) Excess distribution on a logarithmic scale



(b) Tail of the distribution on a double logarithmic scale

Fig. 5. Checking the accuracy of a fitted GPD tail model for threshold $u = 6000$

4 Conclusions

Looking at the rapid deployment of advanced multimedia applications by peer-to-peer overlay networks derived from a mesh-pull architecture and their adaptation to mobile networks, improved design approaches must be developed. To address the related challenges in teletraffic engineering, we have provided the new public source monitoring tool Atheris [1], [8] and recently presented a comprehensive analysis concept [20]. It integrates modeling, measurement, and statistical analysis of peer-to-peer traffic flows at the packet and session levels focusing on the insight gained by a single monitoring point in a P2P overlay network.

In this paper we have extended this approach and investigated improved statistical techniques to characterize the peculiarities of a locally observed peer population in popular P2P overlay networks. Using collected flow data at a single peer, we have shown how Pareto and Generalized Pareto models can be applied to isolate the most dominant part of the feeding population of a peer. Our scheme can be used to support the dynamic adaptation of a peer selection strategy to the real needs of streaming media feeding moving clients (cf. [9], [10]).

We have illustrated our classification approach by real flow data of P2P sessions generated by a mobile BitTorrent client in a WiMAX testbed and a stationary client in a SopCast streaming environment. Regarding the SopCast flows and their exchanged volumes, we have shown that there are strong indications that the dominant and most useful part of the peer population obeys a power law. Its distribution can be described by a Pareto model using the wealth excess transform. Considering BitTorrent flows, a generalized Pareto distribution can be applied successfully as appropriate model.

In this respect our modeling approach supplements the practical findings of Tang et al [26] about the overlay topology of SopCast and theoretical findings of Couto da Silva et al [7]. The latter have shown that a fluid-flow modeling approach can be used to describe the behavior of mesh-based P2P streaming systems like SopCast and PPLive and that bandwidth-aware peer scheduling compares favorably to location-aware and random peer selection schemes. Our statistical analysis discussed here and the previous measurement findings on SopCast [20] confirm da Silva's conclusion [7, p. 452] that the observed hierarchical peer structures are strongly influenced by the access bandwidth. According to our insights super peers have mainly high access bandwidth and an institutional embedding and can thus provide a useful upload service to ordinary peers behind standard access links. The active bandwidth measurement technique implemented by our open source tool Atheris [1] can be used by other investigators to study these relations for new P2P streaming protocols.

In [9] we have further realized that a BitTorrent-like protocol, especially BitTorrent's choking algorithm, is not well adapted to the fluctuating conditions in a mobile network. In this regard our sketched dynamic classification techniques have the potential to cope with an efficiency awareness of a P2P dissemination protocol in such an environment. Considering the control plane of a P2P system at a mobile client side, its realization by the tool RapidStream [2] and the support by new device-to-device protocols of mobile clients, these findings will guide the implementation of an improved middleware concept. It is derived from observation-driven, sound control-theoretical design principles. In future work we plan to support our teletraffic results by a distributed measurement campaign in UMTS and LTE networks (cf. [11]). The latter will guide the required efficient adaptation of peer-to-peer signaling and transport protocols in the overlay network spanning a mobile environment with gigabit links.

In conclusion, we are convinced that our integrated monitoring and analysis approach can provide a deeper insight into the dynamics of P2P multimedia applications and support the rapid development of appropriate teletraffic models

and control algorithms at the packet and session levels regarding the edge link in front of a peer and the structural level of the overlay.

Acknowledgement. The authors express their sincere thanks to P. Eittenberger at Otto-Friedrich-University for his support.

References

1. Atheris, <http://sourceforge.net/projects/atheris/>
2. RapidStream, <http://www.ktr.uni-bamberg.de/project/rapidStream.shtml>
3. Aalto, S., et al.: Segmented P2P Video-on-Demand: Modeling and Performance. In: Proc. 22nd International Teletraffic Congress, Amsterdam, Netherlands (2010)
4. Ali, S., Mathur, A., Zhang, H.: Measurement of commercial peer-to-peer live video streaming. In: Proc. of ICST Workshop on Recent Advances in Peer-to-Peer Streaming, Waterloo, Canada (2006)
5. Aksenov, S.V., Savageau, M.A.: Some properties of the Lerch family of discrete distributions. University of Michigan (February 2008)
6. Castillo, E., Hadi, A.S.: Fitting the Generalized Pareto Distribution to Data. *Journal of the American Statistical Association* 92(440), 1609–1620 (1997)
7. Couto da Silva, A.P., et al.: Chunk distribution in mesh-based large-scale P2P streaming systems: A fluid approach. *IEEE Trans. on Parallel and Distributed Systems* 22(3), 451–463 (2011)
8. Eittenberger, P.M., Krieger, U.R.: Atheris: A First Step Towards a Unified Peer-to-Peer Traffic Measurement Framework. In: 19th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2011, Ayia Napa, Cyprus, February 9–11, pp. 574–581 (2011)
9. Eittenberger, P.M., Kim, S., Krieger, U.R.: Damming the Torrent: Adjusting BitTorrent-like Peer-to-Peer Networks to Mobile and Wireless Environments. *Advances in Electronics and Telecommunications* 2(3), 14–22 (2011)
10. Eittenberger, P.M., Herbst, M., Krieger, U.R.: RapidStream: P2P Streaming on Android. In: Proc. 19th International Packet Video Workshop, PV 2012, Munich, Germany, May 10–11, pp. 125–130 (2012)
11. Eittenberger, P.M., Schneider, K., Krieger, U.R.: Location-aware traffic analysis of a peer-to-peer streaming application in a HSPA network. In: 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2013, Belfast, Northern Ireland, February 27–March 1 (to appear, 2013)
12. Eittenberger, P.M., Krieger, U.R., Markovich, N.M.: Teletraffic modeling of peer-to-peer traffic. In: Laroque, C., et al. (eds.) Proc. Winter Simulation Conference, WSC 2012, Berlin, Germany, December 9–12 (2012)
13. Hei, X., Liang, C., Liang, J., Liu, Y., Ross, K.W.: A measurement study of a large-scale P2P IPTV system. *IEEE Tran. on Multimedia* 9(8), 1672–1687 (2007)
14. Johnson, N.L., Kotz, S.: *Distributions in Statistics - Continuous Univariate Distributions I*. Wiley, New York (1970)
15. Kim, S., et al.: Measurement and analysis of BitTorrent traffic in mobile WiMAX networks. In: 10th IEEE International Conference on Peer-to-Peer Computing, IEEE P2P 2010, pp. 1–4. IEEE (2010)
16. Li, X., Shaked, M.: The observed total time on test and the observed excess wealth. *Statistics & Probability Letters* 68, 247–258 (2004)

17. Liu, F., Li, Z.: A Measurement and Modeling Study of P2P IPTV Applications. In: Proc. of the 2008 International Conference on Computational Intelligence and Security, vol. 1, pp. 114–119 (2008)
18. Markovich, N.M.: Nonparametric Analysis of Univariate heavy-tailed Data. Wiley, Chichester (2007)
19. Markovich, N.M., Krieger, U.R.: Statistical Analysis and Modeling of Skype VoIP Flows. Special Issue Heterogeneous Networks: Traffic Engineering and Performance Evaluation. *Computer Communications* 33, 11–21 (2010)
20. Markovich, N.M., Biernacki, A., Eittenberger, P., Krieger, U.R.: Integrated Measurement and Analysis of Peer-to-Peer Traffic. In: Osipov, E., Kassler, A., Bohnert, T.M., Masip-Bruin, X. (eds.) WWIC 2010. LNCS, vol. 6074, pp. 302–314. Springer, Heidelberg (2010)
21. Newman, M.E.J.: Power laws, Pareto distributions and Zipf’s law. *Contemporary Physics* 46, 323–351 (2005)
22. Peltotalo, J., et al.: Peer-to-peer streaming technology survey. In: Proc. of the Seventh International Conference on Networking, ICN 2008, pp. 342–350. IEEE Computer Society, Washington (2008)
23. Qiu, D., Srikant, R.: Modeling and performance analysis of BitTorrent like peer-to-peer networks. In: Proc. ACM SIGCOMM 2004, pp. 367–378 (2004)
24. Sentinelli, A., Marfia, G., Gerla, M., Kleinrock, L., Tewari, L.: Will IPTV ride the peer-to-peer stream? *IEEE Communications Magazine* 45(6), 86–92 (2007)
25. Silverston, T., et al.: Traffic analysis of peer-to-peer IPTV communities. *Computer Networks* 53(4), 470–484 (2009)
26. Tang, S., Lu, Y., Hernández, J.M., Kuipers, F., Van Mieghem, P.: Topology Dynamics in a P2PTV Network. In: Fratta, L., Schulzrinne, H., Takahashi, Y., Spaniol, O. (eds.) NETWORKING 2009. LNCS, vol. 5550, pp. 326–337. Springer, Heidelberg (2009)
27. Xu, K., et al.: A Model Approach to Estimate Peer-to-Peer Traffic Matrices. In: Proc. IEEE INFOCOM 2011, Shanghai, April 10–15, pp. 676–684 (2011)
28. Zörnig, P., Altmann, G.: Unified representation of Zipf distributions. *Computational Statistics & Data Analysis* 19, 461–473 (1995)

Part II

**Traffic Classification and
Anomaly Detection**

Reviewing Traffic Classification

Silvio Valenti^{1,4}, Dario Rossi¹, Alberto Dainotti^{2,5}, Antonio Pescapè²,
Alessandro Finamore³, and Marco Mellia³

¹ Telecom ParisTech, France

`first.last@enst.fr`

² Università di Napoli Federico II, Italy

`last@unina.it`

³ Politecnico di Torino, Italy

`first.last@polito.it`

⁴ Google, Inc.

⁵ CAIDA, UC San Diego

Abstract. Traffic classification has received increasing attention in the last years. It aims at offering the ability to automatically recognize the application that has generated a given stream of packets from the direct and passive observation of the individual packets, or stream of packets, flowing in the network. This ability is instrumental to a number of activities that are of extreme interest to carriers, Internet service providers and network administrators in general. Indeed, traffic classification is the basic block that is required to enable any traffic management operations, from differentiating traffic pricing and treatment (e.g., policing, shaping, etc.), to security operations (e.g., firewalling, filtering, anomaly detection, etc.).

Up to few years ago, almost any Internet application was using well-known transport layer protocol ports that easily allowed its identification. More recently, the number of applications using random or non-standard ports has dramatically increased (e.g. Skype, BitTorrent, VPNs, etc.). Moreover, often network applications are configured to use well-known protocol ports assigned to other applications (e.g. TCP port 80 originally reserved for Web traffic) attempting to disguise their presence.

For these reasons, and for the importance of correctly classifying traffic flows, novel approaches based respectively on packet inspection, statistical and machine learning techniques, and behavioral methods have been investigated and are becoming standard practice. In this chapter, we discuss the main trend in the field of traffic classification and we describe some of the main proposals of the research community.

We complete this chapter by developing two examples of behavioral classifiers: both use supervised machine learning algorithms for classifications, but each is based on different features to describe the traffic. After presenting them, we compare their performance using a large dataset, showing the benefits and drawback of each approach.

1 Introduction

Traffic classification is the task of associating network traffic with the generating application. Notice that the TCP/IP protocol stack, thanks to a clear repartition between

Table 1. Taxonomy of traffic classification techniques

Approach	Properties exploited	Granularity	Timeliness	Comput. Cost
Port-based	Transport-layer port [49, 50, 53]	Fine grained	First Packet	Lightweight
Deep Packet Inspection	Signatures in payload [44, 50, 60]	Fine grained	First payload	Moderate, access to packet payload
Stochastic Packet Inspection	Statistical properties of payload [26, 30, 37]	Fine grained	After a few packets	High, eventual access to payload of many packets
Statistical	Flow-level properties [38, 45, 50, 58]	Coarse grained	After flow termination	Lightweight
	Packet-level properties [8, 15]	Fine grained	After few packets	Lightweight
Behavioral	Host-level properties [35, 36, 67]	Coarse grained	After flow termination	Lightweight
	Endpoint rate [7, 28]	Fine grained	After a few seconds	Lightweight

layers, is completely agnostic with respect to the application protocol or to the data carried inside packets. This layered structure has been one of the main reasons for the success of the Internet; nevertheless, sometimes network operators, though logically at layer-3, would be happy to know to which application packets belong, in order to better manage their network and to provide additional services to their customers. Traffic classification is also instrumental for all security operations, like filtering unwanted traffic, or triggering alarms in case of an anomaly has been detected.

The information provided by traffic classification is extremely valuable, sometimes fundamental, for quite a few networking operations [38, 42, 46, 52]. For instance, a detailed knowledge of the composition of traffic, as well as the identification of trends in application usage, is required by operators for a better *network design and provisioning*. *Quality of service* (QoS) solutions [58], which prioritize and treat traffic differently according to different criteria, need first to divide the traffic in different classes: identifying the application to which packets belong is crucial when assigning them to a class. In the same way, traffic classification enables differentiated class *charging* or Service Level Agreements (SLA) verification. Finally, some national governments expect ISPs to perform *Lawful Interception* [6] of illegal or critical traffic, thus requiring them to know exactly the type of content transmitted over their networks. Traffic classification represents in fact the first step for activities such as *anomaly detection* for the identification of malicious use of network resources, and for security operation in general, like firewalling and filtering of unwanted traffic [53, 56].

If, on the one hand, the applications of traffic classification are plentiful, on the other hand, the challenges classifiers have to face are not to be outdone. First, they must deal with an increasing amount of traffic as well as equally increasing transmission rates: to cope with such speed and volume, researchers are looking for *lightweight algorithms* with as little computational requirements as possible. The task is further exacerbated by developers of network applications doing whatever in their power to hide traffic and to elude control by operators: traffic encryption and encapsulation of data in other

protocols are just the first two examples that come to mind. Therefore, researchers had to come out with novel and unexpected ways for identifying traffic.

This Chapter is organized as follows. In Section 2, to provide the background of the field, we define a taxonomy in which we highlight the most important contributions in each category along with their most important characteristics. In Section 3 we discuss the state of the art in the field of traffic classification. In Section 4 we describe the most used machine-learning algorithms in the field of traffic classification. In Section 5 we present two antipodean examples of traffic classification approaches, that we directly compare in iSection 6. Section 7 ends the Chapter.

2 Traffic Classification: Basic Concepts and Definitions

The large body of literature about traffic classification [7, 8, 15, 20, 21, 26, 28, 30, 35, 36, 38, 44, 45, 49, 50, 50, 53, 58, 60, 67] is a further evidence of the great interest of the research community towards this topic. In the following, we will present an overview of the different approaches and methodologies that have been proposed by researchers to solve this issue. It is important to underline that this is far from being an attempt to provide a comprehensive list of all papers in this field (which, given their number, would be particularly tedious). Such a detailed reference can be found in a few surveys [38, 52] or in related community websites (e.g., [1]). Our aim is rather to identify the most important research directions so far, as well as the most representative milestone works and findings, to better highlight our contribution to this already deeply investigated subject. Still, despite this huge research effort, the community has not put the last word on traffic classification yet, as a number of challenges and questions still remain open.

To better structure this overview, we divide the classifiers in a few categories according to the information on which they base the classification. This widely accepted categorization, which reflects also the chronological evolution followed by research, is summarized in Tab. 1. The table lists the most important works in each category along with their most relevant characteristics. The most important properties of a traffic classifier, which determine its applicability to different network tasks [19], are:

Granularity. We distinguish between *coarse-grained* algorithms, which recognize only large family of protocols (e.g. P2P vs non P2P, HTTP vs Streaming) and *fine-grained* classifiers, which, instead, try to identify the specific protocol (e.g. BitTorrent vs eDonkey file-sharing), or even the specific application (e.g. PPlive vs SopCast live streaming).

Timeliness. *Early classification* techniques are able to quickly identify the traffic, after a few packets, thus being suited for tasks requiring a prompt reaction (e.g. security). *Late classification* algorithms take longer to collect traffic properties, and in some case they even have to wait for flow termination (i.e., *post mortem* classification): such techniques are indicated for monitoring tasks, such as charging.

Computational cost. The processing power needed to inspect traffic and take the classification decision is an important factor when choosing a classification algorithm. In the context of packet processing, the most expensive operation is usually packet memory access, followed by regular expression matching.

3 State of the Art

In the first days of the Internet, identifying the application associated with some network packets was not an issue whatsoever: protocols were assigned to well-known transport-layer ports by IANA [2]. Therefore, **Port-based classification** [49, 50, 53] simply extracted such value from the packet header and then look it up in the table containing the port-application associations. Unfortunately *Port-based* classification has become largely unreliable [34, 50]. In fact, in order to circumvent control by ISPs, modern applications, especially P2P ones, either use non-standard ports, or pick a random port at startup. Even worse, they hide themselves behind ports of other protocols – this might enable bypassing firewalls as well. While port-based classification may still be reliable for some portion of the traffic [38], nevertheless it will raise undetectable false-positive (e.g., a non-legitimate application hiding beyond well-known port numbers) and false-negative (e.g., a legitimate application running on non-standard ports) classifications.

To overcome this problem, **Payload-based classifiers** [26, 30, 44, 50, 60] were proposed. They inspect the content of packets well beyond the transport layer headers, looking for distinctive hints of an application protocol in packet payloads. We actually split this family of classification algorithms in two subcategories, *Deep packet inspection* (DPI) techniques that try to match a deterministic set of signatures or regular expressions against packet payload, and *Stochastic packet inspection* (SPI), rather looking at the statistical properties of packet content.

DPI has long provided extremely accurate results [50] and has been implemented in several commercial software products as well as in open source projects [4] and in the Linux kernel firewall implementation [3]. The payload of packets is searched for known patterns, keywords or regular expressions which are characteristic of a given protocol: the website of [3] contains a comprehensive lists of well known patterns. Additionally, DPI is often used in intrusion detection systems [53] as a preliminary step to the identification of network anomalies. Besides being extremely accurate, DPI has been proved to be effective from the very first payload packets of a session [5, 54], thus being particularly convenient for early classification.

Despite its numerous advantages, DPI has some significant drawbacks. First the computational cost is generally high, as several accesses to packet memory are needed and memory speed is long known to represent the bottleneck of modern architectures [66]. String and regular expression matching represent an additional cost as well: although there exist several efficient algorithms and data structures for both string matching and regular expression, hardware implementation (e.g. FPGA), ad hoc coprocessors (e.g. DFA) possibly massively parallel (e.g., GPU) are often required to keep up with current transmission speed [41]. These hardware-based approaches have been analyzed and used to improve the performance of machine learning algorithms, traffic classification approaches, and platforms for network security [11, 32, 43, 62, 64, 68] Yet, it is worth noting that while [64] estimate that the amount of GPUs power can process up to 40 Gbps worth of traffic, bottlenecks in the communication subsystem between the main CPU and the GPU crushes the actual performance down to a mere 5.2 Gbps [64]. Similarly, Network Processors [43] and [62] achieve 3.5 Gbps and 6 Gbps of aggregated traffic rate at most. As we will see, statistical classification outperforms these classification rates without requiring special hardware. Another drawback of DPI is that keywords or

patterns usually need to be derived manually by visual inspection of packets, implying a very cumbersome and error prone trial and error process. Last but not least, DPI fails by design in the case of encrypted or obfuscated traffic.

Stochastic packet inspection (SPI) tries to solve some of these issues, for instance by providing methods to automatically compute distinctive patterns for a given protocol. As an example, authors of [44] define Common Substring Graphs (CSG): an efficient data structure to identify a common string pattern in packets. Other works instead directly apply statistical tools to packet payload: authors of [30] directly use the values of the first payload bytes as features for machine learning algorithms; in [26], instead, a Pearson Chi-square test is used to study the randomness of the first payload bytes, to build a model of the syntax of the protocol spoken by the application. Additionally, this last algorithm is able to deal with protocols with partially encrypted payload, such as Skype or P2P-TV applications.

Authors of [37], instead, propose a fast algorithm to calculate the entropy of the first payload bytes, by means of which they are able to identify the type of content: low, medium and high values of the entropy respectively correspond to text, binary and encrypted content. Authors argue that, even if this is a very rough repartition of traffic and moreover some applications are very likely to use all of these kinds of content, nonetheless such information might reveal useful to prioritize some content over the others (e.g. in enterprise environments, binary transfers corresponding to application updates to fix bugs deserve an high priority). Yet, SPI is still greedy in terms of computational resources, requiring several accesses to packet payload, though with simpler operations (i.e., no pattern matching).

While both [26, 37] use entropy-based classification, a notable difference is represented by the fact that in [26] entropy is computed for chunks of data *across* a stream of packets, while [37] computes entropy over chunks *within* the same packet.

Statistical classification [8, 9, 15, 17, 18, 45, 48, 58, 65] is based on the rationale that, being the nature of the services extremely diverse (e.g., Web vs VoIP), so will be the corresponding traffic (e.g., short packets bursts of full-data packets vs long, steady throughput flows composed of small-packets). Such classifiers exploit several flow-level measurements, a.k.a. *features*, to characterize the traffic of the different applications [45, 48, 58]: a comprehensive list of a large number of possible traffic discriminators can be found in the technical report [47]. Finally, to perform the actual classification, statistical classifiers apply data mining techniques to these measurements, in particular machine learning algorithms.

Unlike payload-based techniques, these algorithms are usually very lightweight, as they do not access packet payload and can also leverage information from flow-level monitors such as [12]. Another important advantage is that they can be applied to encrypted traffic, as they simply do not care what the content of packets is. Nevertheless, these benefits are counterbalanced by a decrease in accuracy with respect to DPI techniques, which is why statistical-based algorithms have not evolved to commercial products yet. Still, researchers claim that in the near future operators will be willing to pay the cost of a few errors for a much lighter classification process.

We can further divide this class of algorithms in a few subclasses according to the data mining techniques employed and to the protocol layer of the features used.

Concerning the first criterion, on one hand, unsupervised clustering of traffic flows [45] (e.g., by means of the K-means algorithm) does not require training and allows to group flows with similar features together, possibly identifying novel unexpected behaviors; on the other hand, supervised machine learning techniques [38,65] (e.g., based on Naive Bayes, C4.5 or Support Vector Machines) need to be trained with already classified flows, but are able to provide a precise labeling of traffic. Regarding the protocol layer, we have classifiers employing only flow-level features [48] (e.g., duration, total number of bytes transferred, average packet-size), as opposed to algorithms using packet-level features [8, 15] (e.g., size and direction of the very first packets of a flow). The former ones are usually capable of late (in some cases only *post-mortem*), coarse-grained classification, whereas the latter ones can achieve early, fine-grained classification.

Finally, **Behavioral classification** [35,36,67] moves the point of observation further up in the network stack, and looks at the whole traffic received by a host, or an (IP:port) endpoint, in the network. By the sole examination of the generated traffic patterns (e.g., how many hosts are contacted, with which transport layer protocol, on how many different ports) behavioral classifiers try to identify the application running on the target host. The idea is that different applications generate different patterns: for instance, a P2P host will contact many different peers typically using a single port for each host, whereas a Web server will be contacted by different clients with multiple parallel connections.

Some works [35, 67] characterize the pattern of traffic at different levels of detail (e.g., social, functional and application) and employ heuristics (such as the number of distinct ports contacted, or transport-layer protocols used) to recognize the class of the application running on a host (e.g., P2P vs HTTP). Works taking the behavioral approach to its extreme analyze the graph of connections between endpoints [31, 33], showing that P2P and client-server application generate extremely different connection patterns and graphs. They prove also that such information can be leveraged to classify the traffic of these classes of services even in the network core. A second group of studies [7, 28], instead, propose some clever metrics tailored for a specific target traffic, with the purpose of capturing the most relevant properties of network applications. Combining these metrics with the discriminative power of machine learning algorithms yields extremely promising results. The Abacus classifier [7] belongs to this last family of algorithms, and it is the first algorithm able to provide a fine-grained classification of P2P applications.

Behavioral classifiers have the same advantages of statistical-based classifiers, being lightweight and avoiding access to packet payload, but are usually able to achieve the same accuracy with even less information. Such properties make them the perfect candidate for the most constrained settings. Moreover given the current tendency toward flow-level monitors such as NetFlow [12], the possibility to operate on the sole basis of behavioral characteristics is a very desirable property for classifiers.

We wrap up this overview with an overall consideration on the applicability of classifiers. With few exceptions such as [24], the wide majority of the classification algorithms proposed in literature cannot be directly applied to the network core. Limitations can be either intrinsic to the *methodology* (e.g., behavioral classification typically focuses on endpoint [67] or end-hosts [36] activity), or be tied to the *computational*

complexity (e.g., DPI [26, 44, 50, 60] cannot cope with the tremendous amount of traffic in the network core), or to *state scalability* (e.g., flow-based classification [45, 48] requires to keep a prohibitive amount of per-flow state in the core), or to *path changes* (path instabilities or load balancing techniques can make early classifications techniques such as [8, 15] fail in the core). At the same time, we point out that classifying traffic at the network ingress point is a reasonable choice for ISPs: indeed, traffic can be classified and tagged at the access (e.g., DiffServ IP TOS field, MPLS, etc.), on which basis a differential treatment can then be applied by a simple, stateless and scalable core (e.g., according to the class of application.). We investigate deeper this issue in the second part of this dissertation.

Finally we must deal with a transversal aspect of traffic classification. The heterogeneity of approaches, the lack of a common dataset and of a widely approved methodology, all contribute to make the comparison of classification algorithms a daunting task [59]. In fact, to date, most of the comparison effort has addressed the investigation of different machine learning techniques [8, 23, 65], using the same set of features and the same set of traces. Only recently, a few works have specifically taken into account the comparison problem [10, 38, 42, 52]. The authors of [52] present a qualitative overview of several machine learning based classification algorithms. On the other hand, in [38] the authors compare three different approaches (i.e., based on signatures, flow statistics and host behavior) on the same set of traces, highlighting both advantages and limitations of the examined methods. A similar study is carried also in [42], where authors evaluate spatial and temporal portability of a port-based, a DPI and a flow-based classifier.

4 Machine-Learning Algorithms for Traffic Classification

In this section we will briefly introduce the problem of traffic classification in machine learning theory (with a particular focus on the algorithms we actually employed to exemplify the traffic classification performance in 6), all falling in the category of *supervised classification*.

There is a whole field of research on machine learning theory which is dedicated to supervised classification [40], hence it is not possible to include a complete reference in this chapter. Moreover, instead of improving the classification algorithms themselves, we rather aim at taking advantage of our knowledge of network applications to identify good properties, or features, for their characterization. However, some basic concepts are required to correctly understand how we applied machine learning to traffic classification.

A supervised classification algorithm produces a function f , *the classifier*, able to associate some input data, usually a vector \mathbf{x} of numerical attributes x_i called *features*, to an output value c , the class label, taken from a list C of possible ones. To build such a mapping function, which can be arbitrary complex, the machine learning algorithm needs some examples of already labeled data, the *training set*, i.e. a set of couples (\mathbf{x}, c) from which it *learns* how to classify new data. In our case the features x_i are distinctive properties of the traffic we want to classify, while the class label c is the application associated with such traffic.

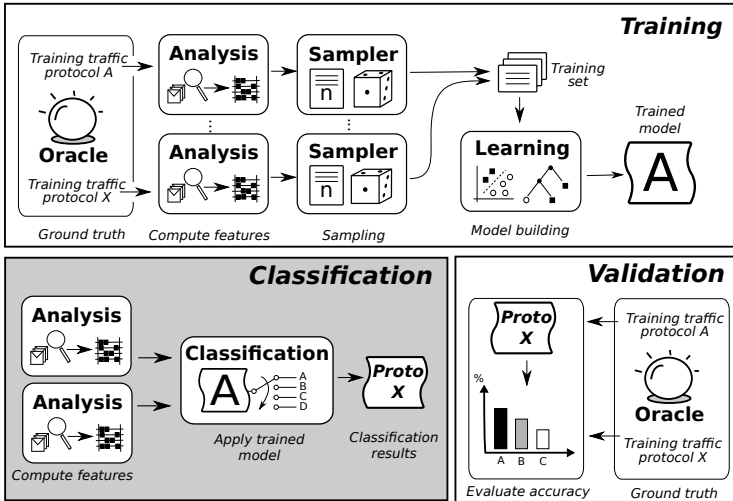


Fig. 1. Common workflow of supervised classification

From a high-level perspective, supervised classification consists of three consecutive phases which are depicted in Fig. 1. During the *training phase* the algorithm is fed with the training set which contains our reference data, the already classified training points. The selection of the training points is a fundamental one, with an important impact on the classifier performance. Extra care must be taken to select enough representative points to allow the classifier to build a meaningful model; however, including too many points is known to degenerate in *overfitting*, where a model is too finely tuned and becomes “picky”, unable to recognize samples which are just slightly different from the training ones.

Notice that, preliminary to the training phase, an *oracle* is used to associate the protocol label with the traffic signatures. Oracle labels are considered accurate, thus representing the *ground truth* of the classification. Finding a reliable ground truth for traffic classification is a research topic on its own, with not trivial technical and privacy issues and was investigated by a few works [16,29].

The second step is the *classification phase*, where we apply the classifier to some new samples, the *test set*, which must be disjoint from the training set. Finally a third phase is needed to *validate* the results, comparing the classifiers outcome against the reference ground truth. This last phase allows to assess the expected performance when deploying the classifier in operational networks.

In this chapter we describe two of the supervised classification algorithms most used in traffic classification literature, namely *Support Vector Machines* and *Classification trees*. This choice is not only based on their large use in the literature of traffic classification, but as they are recognized as having the largest discriminative power in the machine learning community. Specifically, classification accuracy of Support Vector Machines and Classification trees has been compared in [38,61]: Support Vector Machines exhibit the best classification performance in [38], while in [61] the authors

show the superior performance of *Classification trees*. As for the complexity of these approaches,

As for the complexity of these techniques, authors in [22] show how statistical classification based on *Classification trees* can sustain a throughput in excess of 10 Gbps on off-the-shelf hardware, thus outperforming the current state of the art employing GPUs for DPI classification [43, 62, 64]. The next subsections further elaborate the computational complexity of each technique.

4.1 Support Vector Machine

Support Vector Machine (SVM), first proposed by Vapnik [13], is a binary supervised classification algorithm which transforms a non-linear classification problem in a linear one, by means of what is called a “kernel trick”. In the following we intuitively explain how SVM works and refer the reader to [14, 65] for a more formal and complete description of the algorithm.

SVM interprets the training samples as points in a multi-dimensional vector space, whose coordinates are the components of the feature vector \mathbf{x} . Ideally we would like to find a set of surfaces, partitioning this space and perfectly separating points belonging to different classes. However, especially if the problem is non-linear, points might be spread out in the space thus describing extremely complex surface difficult, when not impossible, to find in a reasonable time. The key idea of SVM is then to map, by means of a kernel function, the training points in a newly transformed space, usually with higher or even infinite dimensionality, where points can be separated by the easiest surface possible, an hyperplane. In the target space, SVM must basically solve the optimization problem of finding the hyperplane which (i) separates points belonging to different classes and (ii) has the maximum distance from points of either class. The training samples that fall on the margin and identify the hyperplane are called *Support Vectors* (SV).

At the end of the training phase SVM produces a model, which is made up of the parameters of the kernel function and of a collection of the support vectors describing the partitioning of the target space. During the classification phase, SVM simply classifies new points according to the portion of space they fall into, hence classification is much less computationally expensive than training. Since natively SVM is a binary classifier, some workaround is needed to cope with multi-class classification problems. The strategy often adopted is the *one-versus-one*, where a model for each pair of classes is built and the classification decision is based on a majority voting of all binary models.

Support Vector Machines have proved to be an effective algorithm yielding good performance out-of-the-box without much tuning, especially in complex feature spaces, and has showed particularly good performance in the field of traffic classification [38, 65]. Several kernel functions are available in literature but usually Gaussian kernel exhibits the best accuracy. One drawback of SVM is that models in the multidimensional space cannot be interpreted by human beings and it is not possible to really understand the reason why a model is good or bad. Another, more important, drawback is that the classification process may still require a fair amount of computation. Specifically, the number of operations to be performed is linear in the number of SVs (i.e., the representative

samples) per each class. When the number of classes is large (say, in the order of 100s or 1000s applications), the computational cost can be prohibitive.

4.2 Decision Trees

Decision Trees [39] represent a completely orthogonal approach to the classification problem, using a tree structure to map the observation input to a classification outcome. Again, being this a supervised classification algorithms, we have the same three phases: training, testing and validation.

During the training phase the algorithm builds the tree structure from the sample points: each intermediate node (a.k.a. split node) represents a branch based on the value of one feature, while each leaf represents a classification outcome. The classification process, instead, consists basically in traversing the tree from the root to the leaves with a new sample, choosing the path at each intermediate node according to the criteria individuated by the training phase. Like in SVM, the classification process is way more lightweight than the learning phase. One big advantage of this algorithm over SVM is that the tree can be easily read and eventually interpreted to understand how the algorithms leverages the features for the classification. Another advantage is that classification is based on conditional tests and if-then-else branches, which make it computationally very efficient with respect to SVM.

Literature on this subject contains quite a few decision tree building algorithms, which differ in the way they identify the feature and threshold value for the intermediate split nodes. The best known example of classification tree is the C4.5 algorithm [39], which bases such selection on the notion of *Information Gain*. This is a metric from information theory which measures how much information about the application label is carried by each features, or, in other words, how much the knowledge of a feature tells you about the value of the label. We delay a formal definition of the information gain metric to the next chapter, where we take advantage of it for feature selection purposes. After calculating the information gain of each feature for the training set points, C4.5 picks as splitting feature for each node the one which maximizes such a score: this strategy of using the most helpful attributes at each step is particular efficient, yielding trees of very limited depth (since the most critical split nodes are located toward the top of the tree), which further simplify the computational requirement.

5 Two Antipodean Examples

In this section, we overview a couple of techniques we propose for the online classification of traffic generated by P2P applications (and, possibly, non-P2P application as well).

We mainly consider two approaches with radically different designs. One approach, named Kiss [25, 26], is *payload* based: it inspects the packet payload to automatically gather a stochastic description of the content, thus inferring the *syntax* of the application protocol rather than payload *semantic*. The other approach, named Abacus [7, 63], is instead *behavioral*: it analyzes the transport level exchanges of P2P applications, discriminating between different *protocol dynamics*.

Both Kiss and Abacus achieve very reliable classification but, in reason of their different design, have their pros and cons. For instance, payload-based classification fails when data is fully encrypted (e.g., IPsec, or encrypted TCP exchanges), while the behavioral classifier is unable to classify a single flow (i.e., as protocol dynamics need the observation of multiple flows). A detailed comparison of both techniques is reported in Sec. 6

5.1 Kiss: Stochastic Payload-Based Classification

High-Level Idea. The first approach we consider is based on the analysis of packet payload, trying to detect the syntax of the application protocol, rather than its semantic. The process is better understood by contrasting it with DPI, which typically searches keywords to identify a specific protocol. With a human analogy, this corresponds to trying to recognize the foreign language of an overheard conversation by searching for known words from a small dictionary (e.g., “Thanks” for English language, “Merci” for French, “Grazie” for Italian and so on).

The intuition behind Kiss is that application-layer protocols can however be identified by statistically characterizing the stream of bytes observed in a flow of packets. Kiss automatically builds protocol signatures by measuring entropy (or Chi-Square test) of the packet payload. Considering the previous analogy, this process is like recognizing the foreign language by considering only the cacophony of the conversation, letting the protocol syntax emerge, while discarding its actual semantic.

Fig. 2 reports examples of mean Kiss signatures for popular P2P-TV applications like PPLive, SopCast, TVAnts and Joost that we will use often as examples in this Chapter (and for the comparison in Sec. 6). The picture represents the application layer header, where each group of 4 bits is individually considered: for each group, the amount of entropy is quantified by means of a Chi-Square test χ^2 with respect to the uniform distribution. The syntax of the header is easy to interpret: low χ^2 scores hint to high randomness of the corresponding group of bit, due to obfuscation or encryption; high χ^2 scores instead are characteristic of deterministic fields, such as addresses or identifiers; intermediate values correspond to changing fields, such as counters and flags, or groups of bits that are split across field boundaries. As protocol languages are different, Kiss signatures allow to easily distinguish between applications as emerges from Fig. 2.

Formal Signature Definition. Syntax description is achieved by using a simple Chi-Square like test. The test originally estimates the goodness-of-fit between observed samples of a random variable and a given theoretical distribution. Assume that the possible outcomes of an experiment are K different values. Let O_k be the empirical frequencies of the observed values, out of C total observations ($\sum_k O_k = C$). Let E_k be the number of expected observations of k for the theoretical distribution $E_k = C \cdot p_k$ with p_k the probability of value k . Given that C is large, the distribution of the random variable:

$$X = \sum_{k=1}^K \frac{(O_k - E_k)^2}{E_k} \quad (1)$$

0	4	8	12
1.00	1.00	1.00	1.00
0.99	0.97	1.00	0.99
0.66	0.38	1.00	0.72
1.00	1.00	1.00	1.00
0.52	0.51	0.86	0.83
1.00	0.96	1.00	0.70

(a) Joost

1	4	9	63
1.09	1.09	1.76	1.30
6.11	1.08	1.21	1.21
1.55	1.26	1.21	1.32
1.21	1.37	1.36	1.33
1.03	1.77	1.07	1.07
6.11	1.75	1.77	1.28

(b) SopCast

1	3	4	97
1.09	1.76	1.16	1.13
1.86	1.82	1.13	1.13
1.95	1.16	1.13	1.13
1.13	1.10	1.19	1.17
1.98	1.99	1.98	1.98
1.98	1.99	1.91	1.14

(c) TVAnts

1	6	0	28
1.01	1.99	1.97	1.96
1.03	1.93	1.81	1.22
1.53	1.50	1.91	1.56
1.80	1.26	1.25	1.50
1.53	1.50	1.91	1.53
1.53	1.50	1.59	1.54

(d) PPLive

Fig. 2. Mean Kiss signatures, 24 chunks of 4 bits each (higher value and lighter color correspond to higher determinism)

that represents the distance between the observed empirical and theoretical distributions, can be approximated by a Chi-Square, or χ^2 , distribution with $K - 1$ degrees of freedom. In the classical goodness of fit test, the values of X are compared with the typical values of a Chi-Square distributed random variable: the frequent occurrence of low probability values is interpreted as an indication of a bad fitting. In Kiss, we build a similar experiment analyzing the content of groups of bits taken from the packet payload we want to classify.

Chi-Square signatures are built from *streams* of packets. The first N bytes of each packet payload are divided into G groups of b consecutive bits each; a group g can take integer values in $[0, 2^b - 1]$. From packets of the same stream, we collect, for each group g , the number of observations of each value $i \in [0, 2^b - 1]$; denote it by $O_i^{(g)}$. We then define a window of C packets, in which we compute:

$$X_g = \sum_{i=0}^{2^b-1} \frac{(O_i^{(g)} - E_i^{(g)})^2}{E_i^{(g)}} \quad (2)$$

and collect them in the Kiss signature vector (where, by default, $N = 12$, $G = 24$, $b = 4$, $C = 80$):

$$\bar{X} = [X_1, X_2, \dots, X_G] \quad (3)$$

Once the signatures are computed, one possibility to characterize a given protocol is to estimate the expected distribution $\{E_i^{(g)}\}$ for each group g , so that the set of signatures are created by describing the expected distribution of the protocols of interest in the database. During the classification process then, the observed group g distribution $\{O_i^{(g)}\}$ must be compared to each of the $\{E_i^{(g)}\}$ in the database, for example using the Chi-square test to select the most likely distribution. However, this process ends up in being very complex, since (2) must be computed for each protocol of interest.

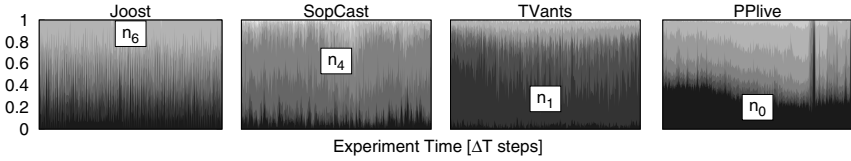


Fig. 3. Temporal evolution of Abacus signatures. Darker color correspond to low order bins, carrying less traffic. Bins are exponential so that $X_i \propto 2^i$, and a mark denotes the most likely bin.

In addition to the high complexity, the comparison with reference distributions fails when the application protocol includes constant values which are randomly extracted for each flow. For example, consider a randomly extracted “flow ID” in a group. Consider two flows, one used for training and one for testing, generated by the same application. Let the training flow packets take the value 12 in that group. Let the test flow packets take instead the value 1 in the same group. Clearly, the comparison of the two observed distributions does not pass the Chi-square test, and the test flow is not correctly classified as using the same protocol as the training flow.

For the above reasons, we propose to simply compare the distance between the observed values and a reference distribution, which we choose as the uniform distribution, i.e., $E_i^{(g)} = E = \frac{C}{2^b}$. In the previous example, the group randomness of the two flows have the same X value, that identify a “constant” field, independently of the actual value of that group. In other terms, we use a Chi-Square like test to measure the randomness of groups of bits, as an implicit estimate of the source entropy.

5.2 Abacus: Fine-Grained Behavioral Classification

High-Level Idea. The Abacus classifier leverages instead on the observation that applications perform different concurrent activities at the same time. Considering for the sake of the example P2P applications, one activity, namely *signaling*, is needed for the maintenance of the P2P infrastructure and is common to all applications. Still, P2P applications differ in the way they actually perform the signaling task, as this is affected by the overlay topology and design (e.g., DHT lookup versus an unstructured flooding search) and by implementation details (e.g., packet size, timers, number of concurrent threads.)

The *data-exchange* activity is instead related to the type of offered service (e.g., file sharing, content, VoIP, VoD, live streaming, etc.). Again, applications are remarkably different, both considering implementation details (e.g., codec, transport layer, neighborhood size, etc.) or the offered service (e.g., low and relatively stable throughput for P2P-VoIP, higher but still relatively stable aggregated incoming throughput for P2P-VoD and TV, largely variable throughput for file-sharing, etc.).

Such difference are so striking, that it is actually possible to finely differentiate between different P2P applications offering the same service: in what follows, we make

an explanatory example on P2P-TV applications. We again consider P2P-TV applications and contrast the possible ways in which they implement the live TV service. Concerning video transfers, for example, some application may prefer to download most of the video content from a few peers, establishing long-lived flows with them, whereas other applications may prefer to download short fixed-sized “chunks” of video from many peers at the same time. Similarly, some application may implement a very aggressive network probing and discovering policy, constantly sending small-size messages to many different peers, while others may simply contact a few super-peers from which they receive information about the P2P overlay. Continuing our human analogy, we may say that some peers will be “shy” and contact a few peers, possibly downloading most of the data from them, while others will be “easy-going” and contact many peers, possibly downloading a few data from each.

These differences are shown in Fig. 3, which depicts the temporal evolution of (a simplified version of) the signature used for traffic classification. To capture the above differences, we assess the shyness of a peer P by gauging the proportion of peers that send to P a given amount of traffic in the range $X_i = [X_i^-, X_i^+]$. We then evaluate an empirical probability mass function p_i (pmf) by normalizing the count n_i of peers sending $x \in X_i$ traffic (e.g., packets or bytes), and by ordering the bins such that $X_{i-i}^+ \leq X_i^-$, i.e. low order bins contain less traffic.

In Fig. 3, darker colors correspond to lower bins, and bins are staggered so that they extend to 1 (due to pmf): for the sake of readability, the most likely (i.e., $\text{argmax}_i n_i$) bin is indicated with a textbox. From Fig. 3, it can be seen that each application has a behavior that, although not stationary over time, is however remarkably different from all the others.

Formal Signature Definition. In the following, we restrict our attention to UDP traffic, although endpoint identification can be extended to applications relying on TCP at the transport layer as well¹. Let us consider the traffic received by an arbitrary end-point $p = (IP, port)$ during an interval of duration ΔT . We evaluate the amount of information received by p simply as the number of received *packets* (although the concept can be extended to the amount of *bytes*, to build more precise signatures [57]).

We partition the space \mathbb{N} of the number of packets sent to p by another peer into $B_n + 1$ bins of exponential-size with base 2: $I_0 = (0, 1]$, $I_i = (2^{i-1}, 2^i]$ for $i = 1, \dots, B_n - 1$ and $I_{B_n} = (2^{B_n-1}, \infty]$. For each ΔT interval, we count the number N_i of peers that sent to p a number of packets $n \in I_i$; i.e., N_0 counts the number of peers that sent exactly 1 packet to p during ΔT ; N_1 the number of peers that sent 2 packets; N_2 the number of peers that sent 3 or 4 packets and, finally, N_{B_n} the number of peers that sent at least $2^{B_n-1} + 1$ packets to p . Let K denote the total

¹ In case TCP is used, the client TCP port is ephemeral, i.e., randomly selected by the Operating System for each TCP connection. The TCP case would require more complex algorithms in case of traffic *generated* from a specific peer, since ephemeral ports differ among flows generated by the same peer. However, the problem vanishes by focusing on the downlink direction: in this case, we aggregate all traffic *received* by a TCP server port, that is the same for all flows of any given peer.

number of peers that contacted p in the interval. The behavioral signature is then defined as $\underline{n} = (n_0, \dots, n_{B_n}) \in \mathbb{R}^{B_n+1}$, where:

$$n_i = \frac{N_i}{\sum_{j=0}^{B_n} N_j} = \frac{N_i}{K} \quad (4)$$

Since \underline{n} has been derived from the pure count of exchanged packets, we name it “Abacus”, which is also a shorthand for “Automated Behavioral Application Classification Using Signatures”. Formally, the signature \underline{n} is the observed probability mass function (pmf) of the number of peers that sent a given number of packets to p in a time interval of duration ΔT (where by default $\Delta T = 5$, $B = 8$).

This function is discretized according to the exponential bins described above. The choice of exponential width bins reduces the size of the signature, while keeping the most significant information that can be provided by the pmf. In fact, as the binning is much finer for short number of packets, short flows with even a small difference in the number of packets are likely to end up (e.g. flows composed by a single packet, two packets and three packets are counted respectively in the component n_0 , n_1 and n_2). On the contrary, longer flows are coarsely grouped together in the higher bins. Intuitively it is more valuable to distinguish between short flows (e.g., distinguishing between single-packet probes versus short signaling exchanges spanning several packets), while there is no gain in having an extreme accuracy when considering long flows (e.g., distinguishing between 500 or 501 packet long flows). This intuition is discussed in [7], where we examine the impact of different binning strategies.

6 Kiss vs. Abacus

At last, we perform a comparison of both approaches, at several levels. To dress a 2π radians view², we consider not only the (i) classification results, but also (ii) functional as well as (iii) complexity aspects. To perform the comparison of the classification results, we consider a common subset of traffic, namely that usual set of P2P-TV applications.

In brief, the algorithms are comparable in terms of accuracy in classifying P2P-TV applications, at least regarding the percentage of correctly classified bytes. Differences instead emerged when we compared the computational cost of the classifiers: with this respect, Abacus outperforms Kiss, because of the simplicity of the features employed to characterize the traffic. Conversely, Kiss is much more general, as it can classify other types of applications as well.

6.1 Methodology

We evaluate the two classifiers on the traffic generated by the common set of P2P-TV applications, namely PPLive, TVAnts, SopCast and Joost. Furthermore we use two distinct sets of traces to assess two different aspects of our classifiers.

² Well, I assume that since “360° degree” is a common saying for everybody, “ 2π radians” should not be an uncommon saying among scientists and engineers.

Table 2. Datasets used for the comparison

Dataset	Duration	Flows	Bytes	Endpoints
Napa-WUT	180 min	73k	7Gb	25k
Operator 2006 (op06)	45 min	785k	4Gb	135k
Operator 2007 (op07)	30 min	319k	2Gb	114k

The first set was gathered during a large-scale active experiment performed in the context of the Napa-Wine European project [51]. For each application we conduct an hour-long experiment where several machines provided by the project partners run the software and captured the generated traffic. The machines involved were carefully configured in such a way that no other interfering application was running on them, so that the traces contain P2P-TV traffic only. This set, available to the research community in [51] is used both to train the classifiers and to evaluate their performance in identifying the different P2P-TV applications.













The second dataset consists of two real-traffic traces collected in 2006 and 2007 on the network of a large Italian ISP. This operator provides its customers with uncontrolled Internet access (i.e., it allows them to run any kind of application, from web browsing to file-sharing), as well as telephony and streaming services over IP. Given the extremely rich set of channels available through the ISP streaming services, customers are not inclined to use P2P-TV applications and actually no such traffic is present in the traces. We verified this by means of a classic DPI classifier as well as by manual inspection of the traces. This set has the purpose of assessing the number of false alarms raised by the classifiers when dealing with non P2P-TV traffic. We report in Tab. 2 the main characteristics of the traces.

To compare the classification results, we employ the *diffinder* tool [55], as already done in [10]. This simple software takes as input the logs from different classifiers with the list of flows and the associated classification outcome. Then, it calculates as output several aggregate metrics, such as the percentage of agreement of the classifiers in terms of both flows and bytes, as well as a detailed list of the differently classified flows enabling further analysis.

6.2 Classification Results

Tab. 3 reports the accuracy achieved by the two classifiers on the test traces using Support Vector Machines (SVM) [14] as learning technique. Each table is organized in a confusion-matrix fashion where rows correspond to real traffic i.e. the expected outcome, while columns report the possible classification results. For each table, the upper part is related to the Napa-Wine traces while the lower part is dedicated to the operator traces. The values in bold on the main diagonal of the tables express the *recall*, a metric commonly used to evaluate classification performance, defined as the ratio of true positives over the sum of true positives and false negatives. The “unknown” column counts the percentage of traffic which was recognized as not being P2P-TV traffic, while the column “not classified” accounts for the percentage of traffic that Kiss cannot classify (as it needs at least $C = 80$ packets for any endpoint).

Table 3. Classification results: Byte-wise confusion matrix for Abacus (left) and Kiss (right)

		Abacus					Kiss					
						un					un	nc
		99.33	-	-	0.11	0.56	99.97	-	-	-	0.01	0.02
		0.01	99.95	-	-	0.04	-	99.96	-	-	0.03	0.01
		0.01	0.09	99.85	0.02	0.03	-	-	99.98	-	0.01	0.01
		-	-	-	99.98	0.02	-	-	-	99.98	0.01	0.01
op06		1.02	-	0.58	0.55	97.85	-	0.07	-	0.08	98.45	1.4
op07		3.03	-	0.71	0.25	96.01	-	0.08	0.74	0.05	96.26	2.87





=PPLive, =Tvants, =Sopcast, =Joost, un=Unknown, nc=not-classified

Table 4. Functional comparison of Abacus and Kiss

Characteristic	Abacus	Kiss
Classification Branch	Behavioral	Stochastic Payload Inspection
Classification Entity	Endpoint	Endpoint/Flow
Input Format	Netflow-like	Packet trace
Target Grain	Fine grained	Fine grained
Protocol Family	P2P-TV	Any
Rejection Criterion	Threshold/Train-based	Train-based
Train-set Size	Big (4000 smp.)	Small (300 smp.)
Time Responsiveness	Deterministic (5sec)	Stochastic (early 80pkts)
Network Deploy	Edge	Edge/Backbone

It is easy to grasp that both the classifiers are extremely accurate, as most of the bytes are correctly classified (flow accuracy is analyzed in [27]). For the Napa-Wine traces the percentage of true positives exceeds 99% for all the considered applications. For the operator traces, again the percentage of true negatives exceeds 96% for all traces, with Kiss showing a overall slightly better performance. These results demonstrate that even an extremely lightweight behavioral classification mechanism, such as the one adopted in Abacus, can achieve the same precision of an accurate payload based classifier.

6.3 Functional Comparison

In the previous section we have shown that the classifiers actually have similar performance for the identification of the target applications as well as the “unknown” traffic. Nevertheless, they are based on very different approaches, both presenting pros and cons, which need to be all carefully taken into account and that are summarized in Tab. 4.

The most important difference is the classification technique used. Even if both classifiers are statistical, they work at different levels and clearly belong to different families of classification algorithms. Abacus is a behavioral classifier since it builds a statistical

representation of the pattern of traffic generated by an endpoint, starting from transport-level data. Conversely, Kiss derives a statistical description of the application protocol by inspecting packet-level data, so it is a payload-based classifier.

The first consequence of this different approach lies in type and volume of information needed for the classification. In particular, Abacus takes as input just a measurement of the traffic rate of the flows directed to an endpoint, in terms of both bytes and packets. Not only this represents an extremely small amount of information, but it could also be gathered by a Netflow monitor, so that no packet trace has to be inspected by the classification engine itself. On the other hand, Kiss must necessarily access packet payload for feature computation: this constitutes a more expensive operation, even if only the first 12 bytes are sufficient to achieve a high classification accuracy.

Despite the different input data, both classifiers work at a fine-grained level, i.e., they can identify the specific application related to each flow and not just the class of applications. This consideration may appear obvious for a payload-based classifier such as Kiss, but it is one of the strength of Abacus over other behavioral classifiers which are usually capable only of a coarse grained classification. Clearly, Abacus pays the simplicity of its approach in terms of possible target traffic, as its classification process relies on some specific properties of P2P traffic. On the contrary, Kiss is more general, it makes no particular assumptions on its target traffic and can be applied to any protocol. Indeed, it successfully classifies not only other P2P applications (e.g., eDonkey Skype, etc.), but traditional client-server applications (e.g., DNS, RTP, etc.) as well.

Another important distinguishing element is the rejection criterion. Abacus defines an hypersphere for each target class and measures the distance of each classified point from the center of the associated hypersphere by means of the Bhattacharyya distance. Then, by employing a threshold-based rejection criterion, a point is label as “unknown” when its distance from the center exceeds a given value. Instead Kiss exploits a multi-class SVM model where all the classes, included the unknown, are represented in the training set. If this approach makes Kiss very flexible, the characterization of the classes can be critical especially for the unknown since it is important that the training set contains samples from all possible protocols other than the target ones.

We also notice that there is an order of magnitude of difference in the size of the training set used by the classifiers. In fact, we trained Abacus with 4000 samples per class (although in some tests we experimented the same classification performance even with smaller training sets) while Kiss needs only about 300 samples per class. On the other hand, Kiss needs at least 80 packets generated from (or directed to) an endpoint in order to classify it. This may seem a strong constraint but [26] actually shows that the percentage of not supported traffic is negligible, at least in terms of bytes.

Finally, for what concerns the network deployment, Abacus needs all the traffic received by the endpoint to characterize its behavior. Therefore, it is only effective when placed at the edge of the network, where all traffic directed to an host transits. Conversely, in the network core Abacus would likely see only a portion of this traffic, so gathering an incomplete representation of an endpoint behavior, which in turn could result in an inaccurate classification. Kiss, instead, is more robust with respect to the deployment position. In fact, by inspecting packet payload, it can operate even on a

Table 5. Computational complexity and resource requirements comparison

	Abacus	Kiss
Memory allocation	$2F$ counters	$2^b G$ counters
Packet processing	<pre> EP_state = hash(IP_d, port_d) FL_state = EP_state.hash(IP_s, port_s) FL_state.pkts ++ FL_state.bytes += pkt_size </pre>	<pre> EP_state = hash(IP_d, port_d) for g = 1 to G do P_g = payload[g] EP_state.0[g][P_g] ++ end for </pre>
<i>Tot. op.</i>	$2 lup + 2 sim$	$(2G+1) lup + G sim$
Feature extraction	<pre> EP_state = hash(IP_d, port_d) for all FL_state in EP_state.hash do p[log2(FL_state.pkts)] += 1 b[log2(FL_state.bytes)] += 1 end for N = count(keys(EP_state.hash)) for all i = 0 to B do p[i] /= N b[i] /= N end for </pre>	<pre> E = C/2^b (precomputed) for g = 1 to G do Chi[g] = 0 for i = 0 to 2^b do Chi[g] += (EP_state.0[g][i]-E)^2 end for Chi[g] /= E end for </pre>
<i>Tot. op.</i>	$(4F+2B+1) lup + 2(F+B) com + 3F sim$	$2^{b+1} G lup + G com + (3 \cdot 2^b + 1) G sim$
Memory allocation	320 bytes	384 bytes
Packet processing	$2 lup + 2 sim$	$49 lup + 24 sim$
Feature extraction	$177 lup + 96 com + 120 sim$	$768 lup + 24 com + 1176 sim$
	Default params: $B=8, F=40$	Default params: $G=24, b=4$

lup=lookup, *com*=complex operation, *sim*=simple operation

limited portion of the traffic generated by an endpoint, provided that the requirement on the minimum number of packets is satisfied.

6.4 Computational Cost

To complete the classifiers comparison, we provide an analysis of the requirements in terms of both memory occupation and computational cost. We calculate these metrics from the formal algorithm specification, so that our evaluation is independent from specific hardware platforms or code optimizations. Tab. 5 compares the costs in a general case, reporting in the bottom portion specific figures for the default parameters.

Memory footprint is mainly related to the data structures used to compute the statistics. Kiss requires a table of $G \cdot 2^b$ counters for each endpoint to collect the observed frequencies employed in the chi-square computation. For the default parameters, i.e.

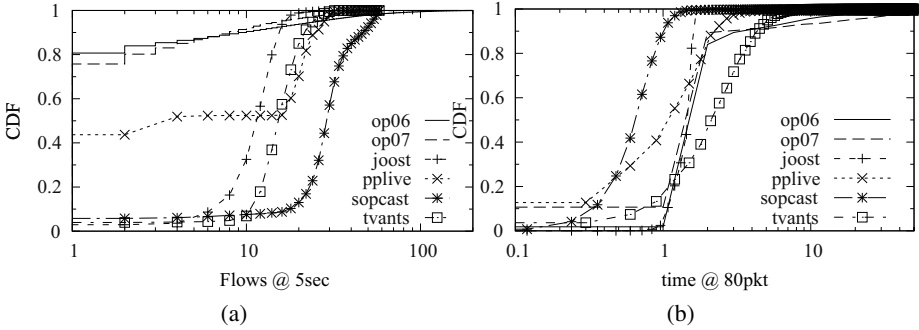


Fig. 4. Cumulative distribution function of (a) number of flows per endpoint and (b) duration of a 80 packet snapshot for the operator traces

$G = 24$ chunks of $b = 4$ bits, each endpoint requires 384 counters. Abacus, instead, requires two counters for each flow related to an endpoint, so the total amount of memory is not fixed but it depends on the number of flows per endpoint. As an example, Fig. 4-(a) reports, for the two operator traces, the CDF of the number of flows seen by each endpoint in consecutive windows of 5 seconds, the default duration of the Abacus time-window. It can be observed that the 90th percentile in the worst case is nearly 40 flows. By using this value as a worst case estimate of the number of flows for a generic endpoint, we can say that $2 \cdot \#Flows = 80$ counters are required for each endpoint. This value is very small compared to Kiss requirements but for a complete comparison we also need to consider the counters dimension. As Kiss uses windows of 80 packets, its counters assume values in the interval $[0, 80]$ so single byte counters are sufficient. Using the default parameters, this means 384 bytes for each endpoint. Instead, the counters of Abacus do not have a specific interval so, using a worst case scenario of 4 bytes for each counter, we can say that 320 bytes are associated to each endpoint. In conclusion, in the worst case, the two classifiers require a comparable amount of memory though on average Abacus requires less memory than Kiss.

Computational cost can be evaluated comparing three tasks: the operations performed on each packet, the operations needed to compute the signatures and the operations needed to classify them. Tab. 5 reports the pseudo code of the first two tasks for both classifiers, specifying also the total amount of operations needed for each task. The operations are divided in three categories and considered separately as they have different costs: *lup* for memory lookup operations, *com* for complex operations (i.e., floating point operations), *sim* for simple operations (i.e., integer operations).

Let us first focus on the packet processing phase, which presents many constraints from a practical point of view, as it should operate at line speed. In this phase, Abacus needs 2 memory lookup operations, to access its internal structures, and 2 integer increments per packet. Kiss, instead, needs $2G + 1 = 49$ lookup operations, half of which are accesses to packet payload. Then, Kiss must compute G integer increments. Since memory read operations are the most time consuming, we can conclude that Abacus should be approximately 20 times faster than Kiss in this phase.

The evaluation of the signature extraction process instead is more complex. First of all, since the number of flows associated to an endpoint is not fixed, the Abacus cost is not deterministic but, like in the memory occupation case, we can consider 40 flows as a worst case scenario. For the lookup operations, Considering $B = 8$, Abacus requires a total of 177 operations, while Kiss needs 768 operations, i.e., nearly four times as many. For the arithmetic operations, Abacus needs 96 floating point and 120 integer operations, while Kiss needs 24 floating point and 1176 integer operations.

Abacus produces signatures every $\Delta T = 5$ seconds, while Kiss signatures are processed every $C = 80$ packets. To estimate the frequency of the Kiss calculation, in Fig. 4(b) we show the CDF of the amount of time needed to collect 80 packets for an endpoint: on average, a new signature is computed every 2 seconds. This means that Kiss calculate feature more frequently than Abacus: i.e., it is more reactive but obviously also more resource consuming.

Finally, the complexity of the classification task depends on the number of features per signature, since both classifiers are based on a SVM decision process. The Kiss signature is composed, by default, of $G = 24$ features, while the Abacus signature contains 16 features: also from this point of view Abacus appears lighter than Kiss.

6.5 Summary of Comparison

We have described, analyzed and compared Kiss and Abacus, two different approaches for the classification of P2P-TV traffic. We provided not only a quantitative evaluation of the algorithm performance by testing them on a common set of traces, but also a more insightful discussion of the differences deriving from the two followed paradigms.

The algorithms prove to be comparable in terms of accuracy in classifying P2P-TV applications, at least regarding the percentage of correctly classified bytes. Differences emerge also when we compared the computational cost of the classifiers. With this respect, Abacus outperforms Kiss, because of the simplicity of the features employed to characterize the traffic. Conversely, Kiss is much more general, as it can classify other types of applications as well.

7 Conclusion

In this Chapter we have reviewed literature in the field of traffic classification, a topic which has increased a lot in relevance during last years. Traffic classification is the building block to enable visibility into the traffic carried by the network, and this it is the key element to empower and implement any traffic management mechanisms: service differentiation, network design and engineering, security, accounting, etc., are all based on the assumption to be able to classify traffic.

Research on Internet traffic classification has produced creative and novel approaches. Yet, as described in this Chapter, there is still room for improvements and contributions in the light of classification techniques and platforms, ground truth, comparison approaches, etc. In particular, the natural evolution of the Internet in which novel applications, protocols and habits are born, proliferate and die, calls for a continuous need to update traffic classification methodologies. This is particular critical considering security aspects in which every bit, byte and packet must be checked.

References

1. CAIDA, The Cooperative Association for Internet Data Analysis, <http://www.caida.org/research/traffic-analysis/classification-overview/>
2. IANA, List of assigned port numbers, <http://www.iana.org/assignments/port-numbers>
3. l7filter, Application layer packet classifier for Linux, <http://l7-filter.clearfoundation.com/>
4. Tstat, <http://tstat.tlc.polito.it>
5. Aceto, G., Dainotti, A., de Donato, W., Pescapè, A.: Portload: Taking the best of two worlds in traffic classification. In: INFOCOM IEEE Conference on Computer Communications Workshops, 15, pp. 1–5 (2010)
6. Bakerand, F., Fosterand, B., Sharp, C.: Cisco Architecture for Lawful Intercept in IP Networks. IETF RFC 3924 (Informational) (October 2004)
7. Bermolen, P., Mellia, M., Meo, M., Rossi, D., Valenti, S.: Abacus: Accurate behavioral classification of P2P-TV traffic. *Elsevier Computer Networks* 55(6), 1394–1411 (2011)
8. Bernaille, L., Teixeira, R., Salamatian, K.: Early application identification. In: Proc. of ACM CoNEXT 2006, Lisboa, PT (December 2006)
9. Carela-Español, V., Barlet-Ros, P., Sole-Simo, M., Dainotti, A., de Donato, W., Pescapè, A.: K-dimensional trees for continuous traffic classification, pp. 141–154 (2010)
10. Cascarano, N., Risso, F., Este, A., Gringoli, F., Salgarelli, L., Finamore, A., Mellia, M.: Comparing P2PTV Traffic Classifiers. In: 2010 IEEE International Conference on Communications (ICC), pp. 1–6 (May 2010)
11. Cascarano, N., Rolando, P., Risso, F., Sisto, R.: Infant: Nfa pattern matching on gpgpu devices. *Computer Communication Review* 40(5), 20–26 (2010)
12. Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954 (Informational) (October 2004)
13. Cortes, C., Vapnik, V.: Support-vector networks. *Machine Learning* 20, 273–297 (1995)
14. Cristianini, N., Shawe-Taylor, J.: An introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, New York (1999)
15. Crotti, M., Dusi, M., Gringoli, F., Salgarelli, L.: Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Computer Communication Review* 37(1), 5–16 (2007)
16. Dainotti, A., de Donato, W., Pescapè, A.: TIE: A Community-Oriented Traffic Classification Platform. In: Papadopoulou, M., Owezarski, P., Pras, A. (eds.) TMA 2009. LNCS, vol. 5537, pp. 64–74. Springer, Heidelberg (2009)
17. Dainotti, A., de Donato, W., Pescapè, A., Salvo Rossi, P.: Classification of network traffic via packet-level hidden markov models 30, 1–5 (2008)
18. Dainotti, A., Pescapè, A., Kim, H.C.: Traffic classification through joint distributions of packet-level statistics. In: GLOBECOM, pp. 1–6 (2011)
19. Dainotti, A., Pescapè, A., Claffy, K.C.: Issues and future directions in traffic classification. *IEEE Network* 26(1), 35–40 (2012)
20. Dainotti, A., Pescapè, A., Sansone, C.: Early Classification of Network Traffic through Multi-classification. In: Domingo-Pascual, J., Shavitt, Y., Uhlig, S. (eds.) TMA 2011. LNCS, vol. 6613, pp. 122–135. Springer, Heidelberg (2011)
21. Dainotti, A., Pescapè, A., Sansone, C., Quintavalle, A.: Using a Behaviour Knowledge Space Approach for Detecting Unknown IP Traffic Flows. In: Sansone, C., Kittler, J., Roli, F. (eds.) MCS 2011. LNCS, vol. 6713, pp. 360–369. Springer, Heidelberg (2011)
22. Santiago del Rfo, P.M., Rossi, D., Gringoli, F., Nava, L., Salgarelli, L., Aracil, J.: Wire-speed statistical classification of network traffic on commodity hardware. In: ACM IMC 2012 (2012)

23. Erman, J., Arlitt, M., Mahanti, A.: Traffic classification using clustering algorithms. In: MineNet 2006: Mining Network Data (MineNet) Workshop at ACM SIGCOMM 2006, Pisa, Italy (2006)
24. Erman, J., Mahanti, A., Arlitt, M., Williamson, C.: Identifying and discriminating between web and peer-to-peer traffic in the network core. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, pp. 883–892 (2007)
25. Finamore, A., Mellia, M., Meo, M., Rossi, D.: KISS: Stochastic Packet Inspection. In: Papadopouli, M., Owezarski, P., Pras, A. (eds.) TMA 2009. LNCS, vol. 5537, pp. 117–125. Springer, Heidelberg (2009)
26. Finamore, A., Mellia, M., Meo, M., Rossi, D.: Kiss: Stochastic packet inspection classifier for udp traffic. *IEEE/ACM Transaction on Networking* 18(5), 1505–1515 (2010)
27. Finamore, A., Meo, M., Rossi, D., Valenti, S.: Kiss to Abacus: A Comparison of P2P-TV Traffic Classifiers. In: Ricciato, F., Mellia, M., Biersack, E. (eds.) TMA 2010. LNCS, vol. 6003, pp. 115–126. Springer, Heidelberg (2010)
28. Fu, T.Z.J., Hu, Y., Shi, X., Chiu, D.M., Lui, J.C.S.: PBS: Periodic Behavioral Spectrum of P2P Applications. In: Moon, S.B., Teixeira, R., Uhlig, S. (eds.) PAM 2009. LNCS, vol. 5448, pp. 155–164. Springer, Heidelberg (2009)
29. Gringoli, F., Salgarelli, L., Dusi, M., Cascarano, N., Risso, F., Claffy, K.C.: GT: picking up the truth from the ground for internet traffic. *ACM SIGCOMM Comput. Commun. Rev.* 39(5), 12–18 (2009)
30. Haffner, P., Sen, S., Spatscheck, O., Wang, D.: ACAS: automated construction of application signatures. In: ACM SIGCOMM Workshop on Mining Network Data (Minenet 2005), Philadelphia, PA (August 2005)
31. Iliofotou, M., Pappu, P., Faloutsos, M., Mitzenmacher, M., Singh, S., Varghese, G.: Network monitoring using traffic dispersion graphs (tdgs). In: Proc. of IMC 2007, San Diego, California, USA (2007)
32. Jamshed, M., Lee, J., Moon, S., Yun, I., Kim, D., Lee, S., Yi, Y., Park, K.S.: Kargus: a highly-scalable software-based intrusion detection system (2012)
33. Jin, Y., Duffield, N., Haffner, P., Sen, S., Zhang, Z.-L.: Inferring applications at the network layer using collective traffic statistics. *SIGMETRICS Perform. Eval. Rev.* 38 (June 2010)
34. Karagiannis, T., Broido, A., Brownlee, N., Klaffy, K.C., Faloutsos, M.: Is P2P dying or just hiding? In: IEEE GLOBECOM 2004, Dallas, Texas, US (2004)
35. Karagiannis, T., Broido, A., Faloutsos, M., Claffy, K.C.: Transport layer identification of P2P traffic. In: 4th ACM SIGCOMM Internet Measurement Conference (IMC 2004), Taormina, IT (October 2004)
36. Karagiannis, T., Papagiannaki, K., Taft, N., Faloutsos, M.: Profiling the End Host. In: Uhlig, S., Papagiannaki, K., Bonaventure, O. (eds.) PAM 2007. LNCS, vol. 4427, pp. 186–196. Springer, Heidelberg (2007)
37. Khakpour, A.R., Liu, A.X.: High-speed flow nature identification. In: Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems, ICDCS (2009)
38. Kim, H., Claffy, K., Fomenkov, M., Barman, D., Faloutsos, M., Lee, K.: Internet traffic classification demystified: myths, caveats, and the best practices. In: Proc. of ACM CoNEXT 2008, Madrid, Spain (2008)
39. Kohavi, R., Quinlan, R.: Decision tree discovery. In: Handbook of Data Mining and Knowledge Discovery, pp. 267–276. University Press (1999)
40. Kotsiantis, S.B.: Supervised machine learning: A review of classification techniques. In: Proceeding of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, pp. 3–24. IOS Press, Amsterdam (2007)

41. Kumar, S., Crowley, P.: Algorithms to accelerate multiple regular expressions matching for deep packet inspection. In: Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM 2006), pp. 339–350 (2006)
42. Li, W., Canini, M., Moore, A.W., Bolla, R.: Efficient application identification and the temporal and spatial stability of classification schema. *Computer Networks* 53(6), 790–809 (2009)
43. Liu, Y., Xu, D., Sun, L., Liu, D.: Accurate traffic classification with multi-threaded processors. In: IEEE International Symposium on Knowledge Acquisition and Modeling Workshop, KAM (2008)
44. Ma, J., Levchenko, K., Kreibich, C., Savage, S., Voelker, G.M.: Unexpected means of protocol inference. In: 6th ACM SIGCOMM Internet Measurement Conference (IMC 2006), Rio de Janeiro, BR (October 2006)
45. McGregor, A., Hall, M., Lorier, P., Brunskill, J.: Flow Clustering Using Machine Learning Techniques. In: Barakat, C., Pratt, I. (eds.) PAM 2004. LNCS, vol. 3015, pp. 205–214. Springer, Heidelberg (2004)
46. Mellia, M., Pescapè, A., Salgarelli, L.: Traffic classification and its applications to modern networks. *Computer Networks* 53(6), 759–760 (2009)
47. Moore, A., Zuev, D., Crogan, M.: Discriminators for use in flow-based classification. Technical report, University of Cambridge (2005)
48. Moore, A.W., Zuev, D.: Internet traffic classification using bayesian analysis techniques. In: ACM SIGMETRICS 2005, Banff, Alberta, Canada (2005)
49. Moore, D., Keys, K., Koga, R., Lagache, E., Claffy, K.C.: The coralreef software suite as a tool for system and network administrators. In: Proceedings of the 15th USENIX Conference on System Administration, San Diego, California (2001)
50. Moore, A.W., Papagiannaki, K.: Toward the Accurate Identification of Network Applications. In: Dovrolis, C. (ed.) PAM 2005. LNCS, vol. 3431, pp. 41–54. Springer, Heidelberg (2005)
51. Napa-Wine, <http://www.napa-wine.eu/>
52. Nguyen, T.T.T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials* 10(4), 56–76 (2008)
53. Paxson, V.: Bro: a system for detecting network intruders in real-time. *Elsevier Comput. Netw.* 31, 2435–2463 (1999)
54. Risso, F., Baldi, M., Morandi, O., Baldini, A., Monclus, P.: Lightweight, payload-based traffic classification: An experimental evaluation. In: Proc. of IEEE ICC 2008 (May 2008)
55. Risso, F., Cascarano, N.: Difffinder, <http://netgroup.polito.it/research-projects/17-traffic-classification>
56. Roesch, M.: Snort - lightweight intrusion detection for networks. In: Proceedings of the 13th USENIX Conference on System Administration, LISA 1999, pp. 229–238. USENIX Association (1999)
57. Rossi, D., Valenti, S.: Fine-grained traffic classification with Netflow data. In: TRAFFIC Analysis and Classification (TRAC) Workshop at IWCMC 2010, Caen, France (June 2010)
58. Roughan, M., Sen, S., Spatscheck, O., Duffield, N.: Class-of-service mapping for QoS: a statistical signature-based approach to IP traffic classification. In: ACM SIGCOMM Internet Measurement Conference (IMC 2004), Taormina, IT (October 2004)
59. Salgarelli, L., Gringoli, F., Karagiannis, T.: Comparing traffic classifiers. *ACM SIGCOMM Comp. Comm. Rev.* 37(3), 65–68 (2007)
60. Sen, S., Spatscheck, O., Wang, D.: Accurate, scalable in-network identification of p2p traffic using application signatures. In: 13th International Conference on World Wide Web (WWW 2004), New York, NY, US (May 2004)
61. Lim, Y.S., Kim, H., Jeong, J., Kim, C.K., Kwon, T.T., Choi, Y.: Internet traffic classification demystified: on the sources of the discriminative power. In: CoNEXT, p. 9 (2010)

62. Szabó, G., Gódor, I., Veres, A., Malomsoky, S., Molnár, S.: Traffic classification over Gbit speed with commodity hardware. *IEEE J. Communications Software and Systems* 5 (2010)
63. Valenti, S., Rossi, D., Meo, M., Mellia, M., Bermolen, P.: Accurate, Fine-Grained Classification of P2P-TV Applications by Simply Counting Packets. In: Papadopouli, M., Owezarski, P., Pras, A. (eds.) *TMA 2009. LNCS*, vol. 5537, pp. 84–92. Springer, Heidelberg (2009)
64. Vasiliadis, G., Polychronakis, M., Ioannidis, S.: Midea: a multi-parallel intrusion detection architecture. In: *ACM Conference on Computer and Communications Security*, pp. 297–308 (2011)
65. Williams, N., Zander, S., Armitage, G.: A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM CCR* 36(5), 5–16 (2006)
66. Wulf, W.A., Mckee, S.A.: Hitting the memory wall: Implications of the obvious. *Computer Architecture News* 23, 20–24 (1995)
67. Xu, K., Zhang, Z.-L., Bhattacharyya, S.: Profiling internet backbone traffic: behavior models and applications. *ACM SIGCOMM Comput. Commun. Rev.* 35(4), 169–180 (2005)
68. Zu, Y., Yang, M., Xu, Z., Wang, L., Tian, X., Peng, K., Dong, Q.: Gpu-based nfa implementation for memory efficient high speed regular expression matching. In: *PPOPP*, pp. 129–140 (2012)

A Methodological Overview on Anomaly Detection

Christian Callegari¹, Angelo Coluccia², Alessandro D'Alconzo³, Wendy Ellens⁴,
Stefano Giordano¹, Michel Mandjes^{5,6,7}, Michele Pagano¹, Teresa Pepe¹,
Fabio Ricciato^{2,3}, and Piotr Żuraniewski^{4,5,8}

¹ University of Pisa, Pisa, Italy

² University of Salento, Lecce, Italy

³ Forschungszentrum Telekommunikation Wien, Vienna, Austria

⁴ TNO, Delft, The Netherlands

⁵ Korteweg-de Vries Institute for Mathematics, University of Amsterdam,
The Netherlands

⁶ EURANDOM, Eindhoven University of Technology, Eindhoven,
The Netherlands

⁷ CWI, Amsterdam, The Netherlands

⁸ AGH University of Science and Technology, Krakow, Poland

1 Introduction

In this Chapter we give an overview of statistical methods for anomaly detection (AD), thereby targeting an audience of practitioners with general knowledge of statistics. We focus on the applicability of the methods by stating and comparing the conditions in which they can be applied and by discussing the parameters that need to be set.

Apart from simply providing the readers with an overview of the different statistical methods that can be applied to AD, it is also the goal of this survey (which is different from the several surveys on the topic already available in the literature - e.g., [94]) to cover two other complementary important aspects:

- a discussion of the operational aspects of AD, analyzing the features of network data as well as providing some guidelines for the design of an AD tool;
- a detailed description of the mathematical insights of one of the methods so as to highlight the theoretical background of the approaches and the links with other important branches of statistics, as well as the complexity behind the development of such methods (including the assumptions under which the methods are developed).

In more detail, the remainder of the Chapter is organized in the following way: the next Section starts by providing an informal definition of the AD problem, then introduces some guidelines and operational issues. The subsequent Section is devoted to the discussion of several statistical methods related to the backbone network scenario, while Section 4 details the changepoint detection approaches from a mathematical point of view. Finally, Section 5 concludes the Chapter with some final remarks.

2 Anomaly Detection in Network Traffic

Anomaly Detection in network traffic is a complex and heterogeneous task. Although diverse approaches are possible, the problem of AD is intrinsically *statistical*, as the key

concept is *expectation* [98]: in fact, an *anomaly* can be defined as anything deviating from an expected behavior. Generally speaking, different techniques can be more or less suitable for a given application scenario according to the exact definition of anomaly, which is strongly dependent on the ultimate goal of the detection task, i.e. what kinds of events are worth reporting to the network operator. Indeed, “anomaly” is an ambiguous concept and may refer either to security problems from external sources (e.g. a DoS attack) or internal issues such as failures, bottlenecks, misconfigurations, etc. In any case, some of these events might be deemed endogenous to the network traffic at hand. Others, conversely, are of great interest for the network operator, in particular events that affect several users at the same time, hence likely point to a network-wide problem that should be addressed and counterbalanced/fixed promptly.

AD in operational networks is a challenging task. Indeed, these are highly heterogeneous, complex and constantly evolving systems, where human errors, equipment malfunctioning and deliberate attacks [80] are non-negligible and mutually interacting events. For cellular networks the scenario is even worse: the functional complexity inherited from the cellular paradigm, coupled with the openness of TCP/IP protocols and the wide heterogeneity of data applications, introduce additional risks and attack models specific for the mobile infrastructure [103,97,80]. Thus, for network operators it is critical to run tools capable to promptly detect emerging anomalies, since these often involve a risk or even a direct damage to the infrastructure and/or the service offered to customers. Indeed, in this context events of particular interest are *large-scale anomalies* — i.e., macro-events that affect many users at the same time — rather than micro-anomalies with impact limited to one or a few users. Since large amount of data needs to be processed — usually collected via a monitoring system or tool — much attention must be paid to complexity and implementation aspects in the design of AD tools.

2.1 How to Deal with Data

There are two main general approaches to handle data gathered from a monitoring system. The simpler one is to aggregate them by summing up the counters in order to obtain scalar time series of traffic variables¹ or related statistics. In this manner, a compact representation of data is made possible. A more complex but also more informative approach is to maintain separate counters for each user, then to analyze the statistical distribution of these counters across users over time. In so doing, one has to deal with multidimensional streams, i.e. sequences of distributions.

The analysis of the whole statistical distribution across users, as opposite to the analysis of scalar aggregates, can grant an intrinsic gain in detection capability. In fact, distributions across users can be built from the set of user-grained counters at different temporal granularity, hence the information remains available in non-aggregated form for statistical processing at multiple timescales. This approach allows to detect anomalies that might not cause appreciable changes in the aggregated traffic, but are visible in the distribution across users, at the cost of larger amount of data to be handled

¹ Examples of traffic variables are the “number of TCP SYN packets sent in uplink to port 80” or the “number of distinct IP addresses contacted”.

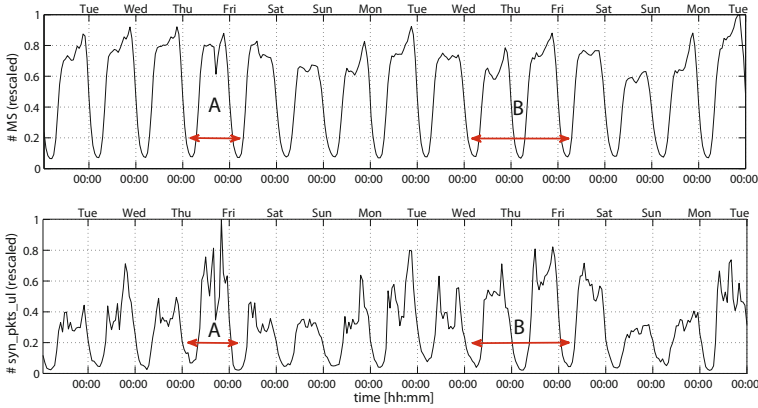


Fig. 1. Number of active users (top) and total volume (bottom) of uplink SYN packets

compared to scalar-based techniques. Furthermore, the distribution-based approach requires a passive monitoring system able to associate each individual packet to a local end-user that sent or received it for uplink and downlink packets respectively. This sophisticated task requires a stateful monitoring based on IP addresses or more unambiguous identifiers like e.g. the subscriber ID in xDSL lines or the IMSI in third-generation (3G) cellular networks, hence has a non-negligible implementation and operation cost.

2.2 Characteristics of Network Traffic

Network traffic exhibits an articulated temporal structure, with marked non-stationarity, daily and weekly pseudo-cycles, physiological pattern changes, etc. A number of in-depth studies, briefly outlined in the following, has been devoted in the last years to these peculiarities (see e.g. [43,56,11,31,79]).

A fundamental characteristic of real network traffic is the dependency of many traffic variables from the *time-of-day*, which is clearly visible in the typical time charts produced by monitoring systems. Fig. 1 represents the number of active users — in this case mobile stations (MS) of a 3G cellular network in Europe — and the overall number of uplink SYN packets at 60 minutes timescale. It is easy to notice that both counters show similar patterns over different days, with some differences on the week-ends that are due to a change in the user behavior compared to working days. Note however that looking at the traffic volumes may be not sufficient to discriminate between physiological and anomalous variations: for instance, referring to Fig. 1 bottom, while the anomalous event “A” exhibits spikes and abrupt changes different from the traffic profile observed in previous days, the long-lasting anomaly “B” does not show clearly these characteristics.

The time-of-day effect is easily explained by observing the typical human activity cycle. In fact, the number of active users and the number of set-up connections follow a slow start in the morning as people wake up and start working, while the slope almost flattens during central day hours. At about 6pm, when most of people finish working,

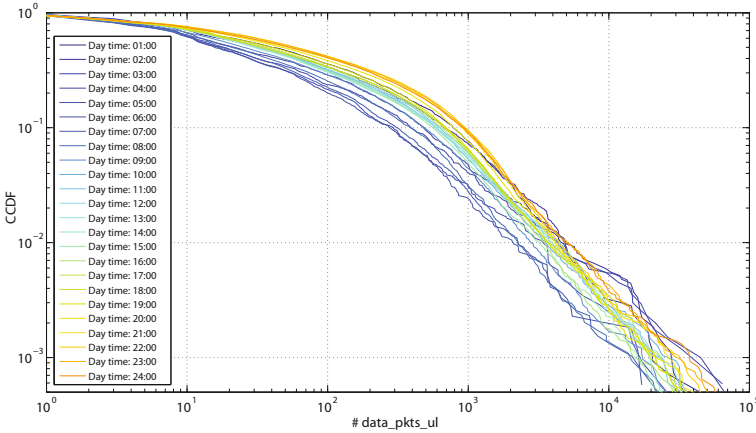


Fig. 2. CCDFs of one entire day for uplink data packets

the traffic starts growing again and reaches a peak at about 10pm. In this period, customers use a different application mix (e.g., file sharing, audio/video streaming, on-line gaming) for their spare-time activities, which results in higher network load. Finally, there is a progressive fade out in the night hours.

This daily profile dependence can be observed at any timescale, with different granularity — the longer the timescale, the smoother the shape of curve. It is important to remark that the time-of-day does not affect only aggregated counters, such as the total traffic volume and number of active users, but also the entire distribution of each traffic variable. In almost all cases a pseudo-cyclical fluctuation of the Complementary Cumulative Distribution Functions (CCDFs) shape takes also place. Such a phenomenon is shown in Fig. 2, for the 1-hour CCDFs of uplink data packets during a whole day. In particular, the shift is clearly visible both at the bulk and at the tail of the distributions.

Distribution changes are due to variations in the traffic composition, caused by changes in the mix of active terminal types (handsets vs. laptops), application mix (see e.g., [91]) and differences in the individual user behavior. For instance, in applications such as web browsing, the coupling between the individual daily/weekly traffic profile and the activity cycles of human users is expected to be stronger than for applications such as file sharing. In fact, the latter can be left unattended active 24/7, while the former is typically bound to the user presence.

Due to peculiarities discussed above, AD in network traffic can benefit from multiresolution analysis, where several traffic variables are processed in parallel over different timescales. The choice of the timescale determines the magnitude of the observable anomalous events. In fact, each event is visible (i.e., produces a deviation) in a limited range, depending on the intensity and duration. Starting from a minimum timebin length (e.g., one minute) data at higher timescales are obtained by integrating observations over time.

The intrinsic non-stationarity of network traffic must be taken into account in the design of an AD system. However, non-stationarity emerges at different timescales

depending on the particular traffic variable under analysis. This means that methods relying on the stationarity assumption can still be successfully applied provided that a suitable timescale is selected. An alternative choice is to adopt techniques that are by design robust against the presence of non-stationarity in the data, which might come at the cost of a somewhat lower accuracy due to the lack of *a priori* information (assumptions) about the underlying process.

2.3 The Detection Problem

The purpose of AD is to reveal significant deviations from a *reference* representative of the “normal” behavior. The adoption of formal statistical processing techniques plays a very important role in providing effective detection methods, and many of these will be reviewed in the following Sections. However, a proper identification of the reference is equally fundamental to achieve good performance.

Even though many approaches can be taken, the AD problem is formalized in general as an abstract hypothesis test. For the null hypothesis \mathcal{H}_0 at timebin t_k , a suitable *reference identification algorithm* must identify a set $\mathcal{S}_0(t_k|t_{k-1}, t_{k-2}, \dots)$ of past observations representing the “normal” behavior. Then, the detection task is to test whether the current traffic distribution $p_{\text{test}}(t_k)$ is consistent with the reference (hypothesis \mathcal{H}_0) or should be considered anomalous (hypothesis \mathcal{H}_1), i.e., in abstract terms:

$$\mathcal{H}_0 : p_{\text{test}} \in \mathcal{S}_0 \quad \text{vs.} \quad \mathcal{H}_1 : p_{\text{test}} \notin \mathcal{S}_0 \quad (1)$$

Network traffic is rich in characteristics that make an accurate statistical modeling difficult, namely non-stationarity and unpredictable fluctuations, but it also contains daily and weekly pseudo-cycles that can be exploited to carry out the AD task. In particular, these pseudo-regularities can be used to build a reference for the expected behavior, as it will be illustrated in Section 2.5.

Unfortunately, these characteristics may change following network reconfigurations and/or modifications in the traffic composition, e.g. due to the spread of a new application or variations in the relative share of active terminal types (smartphones vs. laptops) and application mix [91]. This requires that the AD tool is generally applicable to different kinds of data, and can adapt its parameters to changes in the traffic. Moreover, several traffic variables should be monitored at the same time and analyzed at different granularity in order to reveal events observable at diverse timescales.

2.4 Guidelines for Designing a Network AD Tool

From the discussion above the requirements of an AD tool with operational value can be identified:

- *versatility*, to model different traffic variables at different timescales and aggregation;
- *low-complexity*, to allow on-line implementation for a sufficiently large number of traffic variables and user population;

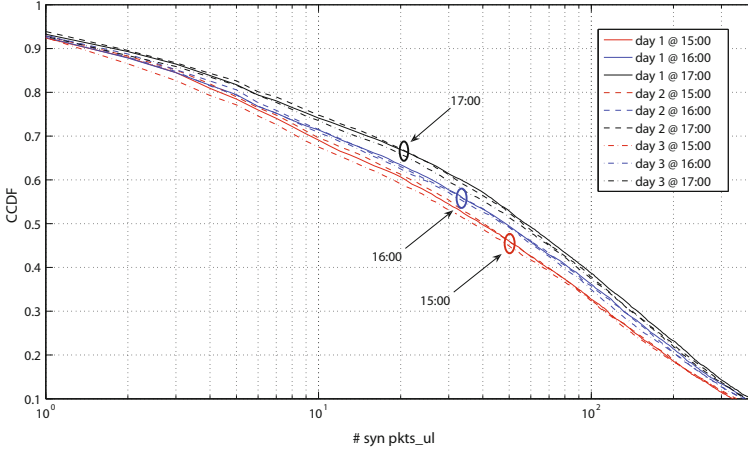


Fig. 3. SYN packets to port 80 for three days at the same hour

- *adaptiveness*, to adapt the detection rules automatically to changes in the network architecture and traffic composition with only minor parameter tuning.

Therefore, the workflow in the design of a network AD tool can be outlined as follows:

- i) definition of a (reliable and versatile) reference identification algorithm;
- ii) derivation of a (low-complexity and general) detection rule;
- iii) identification of (practical) operational criteria for on-line operation.

The reference set identification algorithm i) is usually a suitable heuristic that is designed based on the data at hand. Since it is tailored to the kind of data to be processed, e.g. scalar volume aggregates or conversely time series of distributions, this aspect is often not much addressed in the research literature. In many cases, in fact, the reference set is chosen by means of simple sliding-window approaches, which however might not take into account all the characteristics of real traffic. An illustration of some of these issues will be given in Section 2.5.

Detection rules are conversely a well studied topic, and many of the available techniques will be reviewed in Sections 3 and 4.

Finally, the identification of (practical) criteria for on-line operation is a crucial aspect from a network operator's viewpoint. Indeed, several AD methods exhibit excellent performance in controlled testbeds, but lack the final step to the real network since they miss practical mechanisms for unsupervised on-line operation. In fact, a number of issues must be addressed when deploying theoretical tools in operational contexts, as it will be briefly discussed in Section 2.6.

2.5 Practical Issues in Building a Reference Set

As described in Section 2.4, the preliminary step to AD is to identify a suitable reference set for \mathcal{H}_0 taking into account the traffic peculiarities. The marked non-stationarity

in both aggregate volumes and statistical distributions implies that the reference identification shall be dynamic, i.e. adaptive.

The use of past distributions for building a reference (expected) traffic profile is rooted in the pseudo-cyclical variations of traffic characteristics. Although these are visible also on the aggregate traffic, the analysis of the temporal variations for the whole distribution allows to distinguish between different kinds of traffic changes — in particular, to discriminate between physiological and anomalous variations (recall the discussion about Fig. 1) — hence to be more selective in the detection task. This can be easily realized by looking at Fig. 3, which reports the CCDF of the number of SYN packets to port 80 for three consecutive days at time 15:00, 16:00 and 17:00. The correlations in the distribution shapes at the same hour across the different days can be exploited to identify a reference set representative of the expected “normal” behavior.

Ensuring the quality of the selected set, i.e. its representativeness as reference for the “normal” traffic, is a very important issue: a too loose selection criterion, indeed, may inflate the acceptable range of variations in the traffic distributions thus resulting in a higher probability of false negatives; conversely, an excessively strict criterion makes the detector too sensitive to traffic variations, probably leading to more false alarms.

In designing a reference identification algorithm, a first issue is the inhomogeneity in the number of active users that contributed to a given traffic variable. Therefore, some attention must be paid to avoid comparing samples with very different statistical significance — note that the number of active users can vary across two orders of magnitude during 24 hours. In the distribution-based approach, in particular, a *similarity metric* for distributions is needed to select the (past) distributions that are most representative of the normal behavior, which will ultimately form the reference set \mathcal{S}_0 . This is used to assess the degree of “similarity” between the observation at current time k and selected past distributions.

The task of reference identification is made difficult by the likely presence of undetected anomalies or similar behavioral patterns. Moreover, past observations (samples) that were previously marked as “anomalous” by the detector should be excluded from the reference identification procedure: this introduces a sort of feedback loop, as the output of the detector for past samples impacts the identification of the reference and therefore influences future decisions.

In summary, the reference identification algorithm must be designed to cope with changes in traffic volume as well as distribution over time. As mentioned, a key requirement is that it is versatile and general enough to be applicable to any traffic variable at different timescales. Indeed, traffic variables do share the same structural ingredients, but combined in many different ways: for instance, daily/weekly pseudo-cycles are typically strong in traffic variables linked to human-attended applications, while they may be much weaker in machine-to-machine traffic. Therefore, only a reference identification algorithm that does not rely upon specific traffic characteristics can be applied without modifications to traffic variables with very different structural characteristics and at different timescales, irrespective of the distribution shape and granularity.

2.6 Practical Issues for Online Operation

The importance of a proper characterization/tuning of the detector lies in the load put on the human expert who must interpret the alarms. In fact, the interpretation of the reported alarms (i.e., the diagnostic of its root cause) is often difficult, time-consuming, and sometimes controversial. This general fact is exacerbated in large-scale AD, since in many practical cases the interpretation involves external information, both technical (e.g., knowledge about recent network upgrades) and other ones. For example, the introduction of a new tariff package, or the release of a new client version of a popular application, might cause sudden shifts in the user behavior, and hence in the traffic patterns and distributions. The effect on the traffic is typically hard to anticipate, hence it is harder to foresee which kind of anomaly may arise. Therefore, while a detector can provide the means to recognize the statistical *syntax* of anomalies, interpreting their *semantic* will remain up to the human expert. This means that the penalty associated to false alarms, i.e. alarms caused by statistical fluctuations, is serious. Too many of such alarms would overload the human experts in charge of processing them, and ultimately undermine the practical value of the whole detection system. In other words, the tolerable level of false alarms is in fact very low, and the objective is to tune the sensitivity of the algorithm so as to control the probability of a false alarm.

Besides interpretation, another task left to the human expert is the (occasional) re-initialization of the system, which is unavoidable in a number of cases. For instance, user population may react to higher available capacity due to network upgrade by generating more traffic, hence changing persistently the distribution of a certain number of traffic variables. Since the network will never come back to the previous state (i.e. to the traffic distributions observed before the upgrade) an AD tool may continue to generate alarms indefinitely. This is a typical case of a persistent change in the network traffic, thus the human expert must necessarily reinitialize the detection algorithm — forcing the system to “forget the past” — and start a new training phase. In principle, it is possible to implement automatic reinitialization schemes, e.g. based on some threshold on the length of the alarm run. On the other hand, only a human expert can decide whether the change is a “legitimate” transition to a new equilibrium point, or rather a long-lasting anomaly to be fixed.

Finally, a major network reconfiguration like a migration to a new device consists of a number of smaller interventions over several days and even weeks. During the transition the network traffic is so “stormy” that it prevents the learning of any “normal” behavior, making any detection mechanism practically unusable. As pointed out in longitudinal studies (see e.g. [5]) the real traffic is often rich of anomalies like e.g. bandwidth changes, congestions, external attacks, etc. This is an issue that should be taken into account when designing an AD system as well as when interpreting its alarms, as it practically limits somewhat the proper identification of traffic patterns to be used as reference for “normality”.

3 Statistical Based AD Methods

Since the seminal paper by Denning [35], a rich and vast literature has focused on network AD, including several surveys (such as [95]). It is worth mentioning that AD has

been investigated within different research areas and application domains and many AD techniques are quite general, while others are specific for given applications (see, for instance [20,68], to name only a few). Moreover, even in the framework of network AD the same “statistical tool” can be applied in completely different frameworks (ranging from the packet-by-packet analysis of single connections to coarse traffic aggregates) and with different goals (for instance, the wavelet transform can be used to highlight “discontinuities” or to perform a multiresolution analysis).

Several taxonomies have been proposed in the literature, distinguishing for instance among statistical AD, machine learning and discrete algorithms [95]. As a common feature, all the IDSs involve a preprocessor to extract the selected features from raw traffic data (possibly after a sampling procedure, to mitigate the scalability issue). Indeed, due to the wide range of possible attacks different “features” of the traffic can be considered, taking into account different combinations of Layer 3 and 4 packet header fields in order to define suitable input data.

This Section provides an overview of some of the most promising approaches to network AD and discusses how the different methods can be applied to traffic data, focusing on backbone network scenarios. For sake of brevity, most of the mathematical details are omitted for all the methods described in this Section. Just to give an idea of the theoretical background, the approaches based on changepoint detection are instead discussed in detail in the following Section, highlighting the links with other branches of theoretical and applied mathematics, such as queueing theory and Large Deviation Theory.

In more detail, we start by describing the sketches, a method for efficiently reducing the dimension of the data and then we consider an example of machine learning approaches for AD, namely different clustering algorithms. The subsequent subsection is devoted to discrete algorithms for AD, such as streaming change detection, while the remaining part of Section 3 and the entire Section 4 deal with statistical approaches, such as wavelets, principal component analysis, and changepoint detection methods.

3.1 Sketches

Sketches can not be considered as a detection method, nevertheless we introduce them in this document, given that they can be used as a building block of several AD systems [90,36,5,28,15,13,76,55]. Indeed, the use of sketches corresponds to a random aggregation that “efficiently” reduces the dimension of the data (wrt other deterministic aggregations, such as according to input/output routers [13]); moreover, the use of reversible sketches [83] permits to trace back the flows responsible for the anomalies.

Roughly speaking, sketches are a family of data structures that use the same underlying hashing scheme for summarizing data. They differ in how they update hash buckets and use hashed data to derive estimates. Among the different sketches, the one with the best time and space bounds is the so called Count-Min sketch [28].

In more detail, the sketch data structure is a two-dimensional $d \times w$ array $T[l][j]$, where each row l ($l = 1, \dots, d$) is associated to a given hash function h_l . For instance a typical choice is represented by the use of functions belonging to the 4-universal

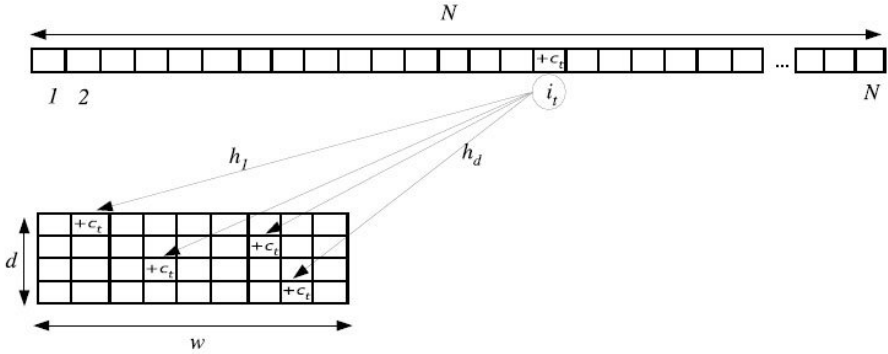


Fig. 4. Sketch: Update Function

hash family² [93]. These functions give an output in the interval $\{1, \dots, w\}$ and these outputs are associated to the columns of the array. As an example, the element $T[l][j]$ is associated to the output value j of the hash function l .

The input data are viewed as a stream that arrives sequentially, item by item. Each item consists of a *hash key*, $i_t \in \{1, \dots, N\}$, and a *weight*, c_t . When new data arrive, the sketch is updated as follows:

$$T[l][h_l(i_t)] = T[l][h_l(i_t)] + c_t \tag{2}$$

The update procedure is realized for all the different hash functions as shown in Fig. 4.

The main advantage of the use of the sketches, is that they permit to process big quantities of data (e.g., quantity of traffic sent/received by each IP address in a network) with big memory savings.

Moreover, given the use of hash functions, it is possible to have some *collisions* in the sketch table, allowing to randomly aggregate the data.

As an example, consider the case of a system that analyzes the quantity of traffic received by each IP address. Having collisions implies that each traffic flow will be part of several random aggregates, each of which will be analyzed to check if it presents any anomaly. This means that, in practice, any flow will be checked more than once, thus, it will be easier to detect an anomalous flow. Indeed it could be masked in a given traffic aggregate, while being detectable in another one [13].

Since sketches are used for AD purposes in conjunction with other algorithms (namely, Stream change detection, Wavelets and Principal component analysis), the overview of the related literature is carried out in the corresponding subsections.

² A class of hash functions $H : \{1, \dots, N\} \rightarrow \{1, \dots, w\}$ is a *k-universal hash* if for any distinct $x_0, \dots, x_{k-1} \in \{1, \dots, N\}$ and any possibly $v_0, \dots, v_{k-1} \in \{1, \dots, w\}$:

$$Pr\{h(x_i) = v_i; \forall i \in \{1, \dots, k\}\} = \frac{1}{w^k}$$

3.2 Clustering

One of the most “classical” approaches, in the field of network AD, is represented by clustering [75][77][50].

The premise is that measurements that correspond to normal operations are similar and thus cluster together. Measurements that do not fall in the normal clusters are seen as *outliers* and thus considered as anomalies. Some of the earlier research in AD evolved from statistical outlier-detection techniques. A survey of various outlier-detection techniques from statistical techniques to machine learning approaches can be found in [46].

The main advantage that clustering provides is the ability to learn from and detect intrusions in the audit data, while not requiring the system administrator to provide explicit descriptions of various attack classes/types. The capability of finding hidden structure in unlabeled data is known in the framework of machine learning as “unsupervised learning”; this feature is also shared by other approaches described in this overview, such as Principal component analysis.

The training data contain both normal and anomalous data. In its purest form, unsupervised AD schemes use unlabeled data for training. Eskin et al. provided a geometric framework to realize an unsupervised AD scheme [39]. In their work, raw input data are mapped to a feature space. They labeled points that are in sparse regions of the feature space as anomalies. They used three different algorithms: a fixed-width clustering-based algorithm, an optimized K-nearest-neighbor algorithm, and an unsupervised variant of the Support Vector Machine (SVM) algorithm. Oldmeadow et al. did further work on the clustering method used in [39] and obtained an improvement in accuracy when the clusters are adaptive to changing traffic patterns [72]. The main drawback of both these works is that the proposed systems are only tested over the KDD 99 dataset [2], derived from the DARPA project [1], that is known to be inadequate to test AD systems [9].

Most of the work devoted to the use of clustering techniques for AD purposes is based on one of the simplest distance-based clustering algorithm, namely the K-means algorithm [70] (or its improved version the fuzzy K-means [37]).

One key point of the distance-based clustering algorithms is the choice of the “similarity” measure used for distinguishing normal from anomalous clusters. Euclidian distance is a commonly-used similarity measure for network measurements, but it has been shown that it is not able to correctly express the distance between two samples when the features are not “uniform”, which is often the case in network AD (features such as inter-arrival time and packet length cannot be considered uniform). Thus, the use of alternate distance (e.g., Mahalanobis distance, weighted Euclidean distance) has been taken into account [57].

In any case, the use of such algorithms suffers from the need to know *a priori* the parameter K , representing the number of the “normal” clusters. Thus, the application of these methods is often unfeasible.

To overcome this limitation, several techniques have been proposed to choose the optimal value of the parameter K [3] or to substitute the K-means algorithm with a density-based clustering technique, such as the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm [41].

The DBSCAN algorithm is at the basis of the Minnesota Intrusion Detection System (MINDS) [38]. The MINDS AD module assigns a degree of outlieriness to each data

point, which is called the local outlier factor (LOF) [7]. The LOF takes into consideration the density of the neighborhood around the observation point to determine its outlierness. In this scheme, outliers are objects that tend to have high LOF values. The advantage of the LOF algorithm is its ability to detect all forms of outliers, including those that cannot be detected by the distance-based algorithms.

As a general consideration, the main advantage provided by the use of clustering algorithms for AD purposes is represented by the fact that these algorithms are usually low demanding from a computational point of view.

3.3 Streaming Change Detection

When referring to Streaming techniques, we describe the network data as streaming data, by using, as an example, the most general model: the Turnstile Model [71].

According to this model, the input data are viewed as a stream that arrives sequentially, item by item. Let $I = \sigma_1, \sigma_2, \dots, \sigma_n$ be the input stream.

Each item $\sigma_t = (i_t, c_t)$ consists of a *key*, $i_t \in \{1, \dots, N\}$, and a *weight*, c_t . The arrival of a new data item causes the update of an underlying function $U_{i_t}(t) = U_{i_t}(t - 1) + c_t$, which represents the sum of the *weights* of a given *key* i_t until time t .

Given the underlying function $U_i[t]$ for all the keys of the stream, we can define the total sum $S(t)$, at step t , as follows:

$$S(t) = \sum_i U_i[t] \quad (3)$$

This model is very general and can be used in quite different scenarios. As an example, in the context of network AD, the key can be defined using one or more fields of the packet header (IP addresses, L4 ports), or entities (like network prefixes or AS number) to achieve a higher level of aggregation, while the underlying function can be the total number of bytes or packets in a flow.

Given such data model, the AD problem can be formulated as a Heavy Hitter (HH) detection problem or a Heavy Change (HC) detection problem. In the HH detection problem, the goal is to identify the set of flows that represent a significantly large portion of the ongoing traffic or of the capacity of the link. In the HC detection problem, the goal is to detect the set of flows that have a drastic change from one time period to another.

Heavy Hitter Detection. A HH, in a dataset, is an element whose relative frequency exceeds a specified threshold.

In more detail, given an input stream $I = \{(i_t, c_t)\}$ with the associated total sum S , a HH is a key, whose associated underlying function U_i is not smaller than a specified portion of the expected size of the whole dataset. The problem can be formalized as follows.

Given a threshold ξ ($0 < \xi < 1$) the set of HHs is defined as:

$$HH = \{i \mid U_i > \xi \cdot S\} \quad (4)$$

In the context of network AD, a HH is an entity which accounts for at least a specified portion of the total activity measured in terms of number of packets, bytes, connections, etc. A HH could correspond to an individual flow or connection. It could also be an aggregation of multiple flows/connections that share some common property, but which themselves may not be HH.

Given this, we define the HH detection problem as the problem of finding all the HHs, and their associated values, in a data stream. As an example, let us consider that the destination IP address is the *key*, and the byte count the *weight*; then in this case the corresponding HH detection problem is to find all the destination IP addresses that account for at least a portion ξ of the total traffic.

The general problem of finding HHs in data streams has been studied extensively in [21], [51], and [26]. The algorithms presented in these papers maintain summary structures that allow element frequencies to be estimated, within a pre-specified error bound; the algorithms differ in whether they make deterministic or probabilistic guarantees on the error bound, and whether they operate over insert-only streams or streams where elements can be inserted and also deleted.

The connection of these algorithms to the computer networks field was first made in [40], [66], [29]. In [40], Estan et al. initiate a new direction in traffic measurement by recommending concentrating on large flows only, i.e., flows whose volumes are above certain thresholds. The authors also propose two algorithms for detecting large flows: Sample and Hold algorithm and Multistage Filters algorithm. Theoretical results show that the errors of these two algorithms are inversely proportional to the memory available. Furthermore the authors note that network measurement problems bear a lot of similarities to measurement problems in other research areas such as data mining, and thus initiate the application of data streaming techniques to network AD. In [66], the authors propose the Sticky Sampling algorithm and the Lossy Counting algorithm to compute approximate frequency counts of elements in a data stream. The proposed algorithms can provide guaranteed error bounds and require small memory. Thus, these algorithms also provide powerful tools for solving the HH detection problem in high-speed networks. In [29], Cormode et al. introduce the Count-Min Sketch method (see Section 3.1) to HH and hierarchical HH detection. The authors note that it is an open problem to develop extremely simple and practical sketches for data streaming applications.

In all these works, the application of such techniques to network AD is not explicitly discussed; only in [24], the authors extend the work presented in [29], by stating that the hierarchical HH detection can be applied to the AD field, without working out the details.

Heavy Change Detection. In the context of AD, the goal of HC detection is to efficiently identify the set of flows that have a drastic change from one time period to another with small memory requirements and limited state information.

Modelling the data stream using the Turnstile model, the problem can be stated as follows: “A HC is a key i , whose associated underlying function U_i , evaluated in a given timebin, significantly differs in size if compared to the same function evaluated in the previous timebins.”

For sake of simplicity, let us suppose that we want to detect the HC related to two adjacent timebins. In this case, a key is a HC if the difference between the values in the two timebins exceeds a given threshold (ψ). The problem can be formalized as follows. Let $U_i[t_1]$ and $U_i[t_2]$ be the values associated to the key i , evaluated in the timebin t_1 and t_2 respectively, and let D_i be the relative difference, defined as $D_i = |U_i[t_1] - U_i[t_2]|$. Then the set of HCs is defined as follows:

$$HC = \{i \mid D_i > \psi\} \quad (5)$$

As an example, in the context of network AD, the goal of HC detection can be to identify the flows that have significant changes in traffic volume from one time period to another.

HC detection has been extensively studied as an important “component” of statistics based IDS. Thus, several works have faced the problem, by proposing, among the others, techniques based on the use of neural networks [42], Markov models [104], and clustering algorithms [96]. Unfortunately, most of the existing HC detection techniques can typically handle a relatively small number of time series. Given today’s traffic volume, directly applying existing techniques on a per-flow basis cannot scale up to the needs of such massive data streams. In this context, sketches have shown great potential. In [90] the authors first apply sketch to the HC detection problem. The input data are summarized using k -ary sketches and then different time series forecast models are implemented on top of the aggregate. The forecast errors are used to identify whether there are significant changes in the stream. In [27] the authors introduce the concept of deltoid for HC detection, where a deltoid is defined as an item that has a large variability. The authors propose a framework based on a structure of Combinational Group Testing to find significant deltoids in high speed networks. Finally, in [17] [18] the authors extend these previous works, by first proposing the idea of detecting HC in the distribution of the HH in the network traffic, without any need of reconstructing the original time series.

3.4 Wavelets

In the last years several works have been devoted to the application of the wavelet transform to the problem of AD.

Although the first example of wavelets dates back to the beginning of the twentieth century (Haar basis), the concept has been “rediscovered” in 1981 by the geophysicist Jean Morlet and the word “wavelet” has been introduced in the scientific literature only in 1984 by the physicist Alex Grossman [99]. The key results for the practical applicability of wavelets to signal processing are represented by the discovery of orthonormal bases with compact support [32] and the introduction of the fast (discrete) wavelet transform [63] in the framework of multiresolution analysis. Along with the introduction of orthonormal bases, the deeper understanding of the wavelet theory has led to a renewed interests also for redundant frames thanks to the additional flexibility in the choice of the mother wavelet.

Roughly speaking, the attractive features of wavelet transform are its localization properties in time and frequency (with the possibility of analyzing the signal at different

timescales without losing time information), the ability of detecting sharp changes in the signals and the low correlation among details coefficients. All these features make the wavelet transform a suitable tool in signal processing application, including AD.

The wavelet decomposition [33] is based on the representation of any finite-energy signal $x(t) \in L^2(\mathbb{R})$ by means of its inner products with a family of functions

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right)$$

where ψ is a suitable function called *mother wavelet*. The parameters a and b may assume real values, leading (under very mild assumptions on ψ) to a redundant signal representation known as *Continuous Wavelet Transform* (CWT). Starting from the CWT, it was later introduced the *Discrete Wavelet Transform* (DWT), restricting a and b to a finite discrete set of values (namely, $a = a_0^m$ and $b = nb_0 a_0^m$, where $m, n \in \mathbb{Z}$ and $a_0 \neq 1$ — for convenience usually $a_0 > 1$ and $b_0 > 0$) and imposing stronger limitations on the mother wavelet.

Under quite stringent constraints on its choice (see, for instance, the well-known Daubechies bases family of compactly-supported mother wavelets, introduced by the Belgian mathematician Ingrid Daubechies in 1988), the functions

$$\psi_{m,n}(t) = a_0^{-\frac{m}{2}} \psi\left(\frac{t - nb_0 a_0^m}{a_0^m}\right) = a_0^{-\frac{m}{2}} \psi(a^{-m}t - nb_0) \quad m, n \in \mathbb{Z}$$

may define an orthonormal dyadic wavelet basis (corresponding to $a_0 = 2$ and $b_0 = 1$).

In the framework of computer networks, wavelets has been introduced as a powerful tool to investigate the self-similar nature [58] of traffic flows and to estimate the Hurst parameter (see, for instance, [82] and references therein). Later, wavelets emerged as a natural tool to monitor and infer network properties (in conjunction with on-line algorithms and careful network experimentation), but the approach described in [45] has mainly a theoretical interest: indeed, as stated by the author, the proposed tools work only with information local to a network and it is impossible to incorporate any knowledge of network topology into these local measurements. A further step in the applicability of wavelets for intrusion detection is [48], where the authors explored how much information about the characteristics of end-to-end network paths can be inferred (by exploiting wavelet-based analysis techniques) relying solely on passive packet-level traces of existing traffic, collected from a single tap point in the network. In more detail, WIND (Wavelet-based INference for Detecting network performance problems) enables an on-line almost real time wavelet-based analysis of measured traffic destined for the busiest subnets. At the end of each observation period (the default value is 10 minutes), WIND generates various statistics and some heuristics are introduced to automate the process of identifying “interesting” periods are introduced.

In [4], a portable platform named IMAPIT (Integrated Measurement Analysis Platform for Internet Traffic) is introduced. The goal of the work is to carry out a detailed signal analysis of network traffic anomalies, by applying general wavelet filters to traffic measurements (data were collected at 5 minute intervals, thereby precluding analysis on finer timescales). The wavelet decomposition is performed using a redundant wavelet system, in order to construct relatively short filters with very good frequency localization.

From a given string of Internet measurements, three output signals are derived, corresponding to Low, Mid and High frequencies. These signals are separately processed, in order to find anomalies (flash-crowds, outages, and DoS attacks) of different duration.

Kim et al. [52] propose a technique for traffic AD through analyzing correlation of destination IP addresses in outgoing traffic at an egress router. They hypothesize that the destination IP addresses have a high correlation degree and the changes in the correlation can be used to identify network traffic anomalies. In order to improve the performance of the decision block, the wavelet decomposition (using Daubechies-6 basis) is performed and “detail coefficients” at proper wavelet levels (which can be chosen according to the timescales of interest) are combined. In case of ambient traffic (free of attacks), the reconstructed signal can be considered as Gaussian noise and the detector needs to be trained for a short time on attack-free traffic traces to derive the anomaly thresholds.

In [62] the basic idea is to combine the wavelet approximation with system identification theory. In more detail, 15 network flow-based features are selected and normal daily traffic is modeled and represented by a set of wavelet approximation coefficients, which can be predicted using an ARX (AutoRegressive with eXogenous) model. Then, the deviation of current input signal from normal/regular behavioral signals is used as input to an outliers detection algorithm based on a Gaussian Mixture Model.

The CWT is used in [30], where the authors propose a cascade architecture made of two different systems: the first one (Rough Detection) is based on classical AD techniques for time series, while the second one (Fine Detection) exploits the CWT redundancy in terms of available scales and coefficients to improve the hits/false alarms trade-off.

Since the Wavelet Transform is basically a powerful signal processing tool, it can be used to enhance the performance of the other algorithms described in this Section; for instance, in [15], [16], [76] and [36] the authors propose different methods based on the combined use of sketches and wavelet analysis. In [19] the “edge-detection” features of wavelets are used in order to improve the detection efficiency of a modified version (with a correction term and a low-pass filtering of the input data) of the CUSUM algorithm (see Section 4.2). The rationale behind the use of wavelets (the Haar basis is considered in the paper) is the fact that, by appropriate selection of the scales of interest, the energy of the changepoint’s signal is concentrated in a few coefficients, permitting to realize a selective filtering of the noise.

Finally, it is worth mentioning a couple of papers in which Wavelet Packets [25], a generalization of Wavelets, are used. In more detail, in [14] the idea is to take advantage of the flexibility of Wavelet Packets in selecting the *most significant* transformed coefficients, for instance through a dynamic selection of the best basis (according to different cost criteria) and the analysis of the corresponding transformed coefficients. Chang et al., in [44], propose a new network AD method based on wavelet packet transform, which can adjust the decomposition process adaptively, and thus improving the detection capability on the middle and high frequency anomalies that otherwise cannot be detected by multiresolution analysis.

3.5 Principal Component Analysis

In recent years, Principal Component Analysis (PCA) has emerged as a very promising technique for detecting a wide variety of network anomalies.

The PCA is a linear transformation that maps a coordinate space onto a new coordinate system whose axes, called Principal Components (PCs), have the property to point in the direction of maximum variance of the residual data (i.e., the difference between the original data and the data mapped onto the previous PCs).

In more detail, the first PC captures the greatest degree of data variance in a single direction, the second one captures the greatest degree of variance of data in the remaining orthogonal directions, and so on.

Thus, the PCs are ordered by the amount of data variance they capture. Typically, the first PCs contribute most of the variance in the original dataset, so that we can describe them with only these PCs, neglecting the others, with minimal loss of variance.

In mathematical terms, to calculate the PCs is equivalent to compute the eigenvectors (see also [86] for a general discussion on the possible interpretation of PCA decomposition). Thus, given the matrix of data $B = \{B_{i,j}\}$, with $1 < i < m$ and $1 < j < t$ (a dataset of m samples captured in t timebins), each PC, v_i , is the i -th eigenvector computed from the spectral decomposition of $B^T B$, that is:

$$B^T B v_i = \lambda_i v_i \quad i = 1, \dots, m \quad (6)$$

where λ_i is the “ordered” eigenvalue corresponding to the eigenvector v_i .

In practice, the first PC, v_1 , is computed as follows:

$$v_1 = \arg \max_{\|v\|=1} \|Bv\| \quad (7)$$

Proceeding recursively, once the first $k - 1$ PCs have been determined, the k -th PC can be evaluate as follows:

$$v_k = \arg \max_{\|v\|=1} \left\| \left(B - \sum_{i=1}^{k-1} B v_i v_i^T \right) v \right\| \quad (8)$$

where $\|\cdot\|$ denotes the L^2 norm.

Once the PCs have been computed, given a set of data and its associated coordinate space, we can perform a data transformation by projecting them onto the new axis. In more detail, given the properties of the transformation we can project the data onto a normal subspace (given by the first significant PCs) and consider the residuals to evaluate how much anomalous the data are.

Originally applied in the framework of image processing, in the last years PCA has been widely used in the domain of traffic AD to solve the problem of high dimensionality of typical IDS datasets, with promising initial results.

In [87] the author first proposed an AD scheme based on PCA. In this work, the method based PCA is compared with a more “classical” approach based on clustering and Local Outlier Factor (LOF).

The empirical bases for the application of PCA to the AD field are provided in [56], where the authors perform an analysis of several traffic measurements taken over two

backbone networks (Abilene and Geant) highlighting, by means of PCA, that these measurements have a small intrinsic dimension. This conclusion allows the authors to think that PCA could be suitable for AD.

Starting from this previous work, the same authors, in [53], introduce the subspace method that allows the separation of a high dimensional space occupied by a set of Network traffic measurements into disjoint subspaces, as representative of the normal and anomalous components of the original data. The authors also define a method for revealing the anomalies and for pinpointing the traffic aggregates responsible for such anomalies.

In [54], the subspace method is then applied to three different metrics (packet count, byte count, and IP-flow count) for the detection of different kinds of anomalies.

A step forward in the method is given in [55], where the previous traffic volume metrics are substituted by the entropy of the empirical distribution of the traffic features, enabling the detection of a larger set of anomalies. The authors also use sketches for aggregating the data.

A recent work by Ringberg et al. [81] applies the method described in [55], highlighting some intrinsic difficulties in tuning the method parameters and the caused system instability.

Other notable techniques that employ the principal component analysis methodology include the works done by Wang et al. [101], Bouzida et al. [6] and Wang et al. [102].

Finally, two recent papers [22] [12] have extended the method, so as to improve the detection performance of the system. The first [22] has introduced a multi metric multi link analysis, which implies to analyze several traffic features taken on several links at the same time. Instead, in [12] the authors modify the “standard” method by introducing a multi timescale approach, making the system able to detect both sudden anomalies (e.g. bursty anomalies) and “slow” anomalies (e.g. increasing rate anomalies).

PCA-based AD techniques appear to be suitable for working on the top of a distributed environment. In this framework, some preliminary studies are presented in [60], where the authors propose a simple distributed architecture for AD based on the use of sketch and PCA.

4 Changepoint Detection Techniques

This Section is about changepoint detection. A *changepoint* in a time series (a dataset with a time component) is a moment at which the underlying probability distribution changes. This means that we focus on data that evolve over time, showing sudden (sharp) and insistent changes; the methods described may be less suitable for detecting gradual changes or outliers, for which we refer the reader to the previous Section. We assume that the data are observed sequentially, i.e., we analyze the observations up to this moment, in every time step adding one more observation. At each time we ask ourselves, given the set of observations until that moment, whether a changepoint has occurred or not. In this way we hope to detect a potential change as soon as possible. Statistically speaking, changepoint detection methods perform a hypothesis test for every time step. In general, they monitor some *test statistic* which is based on the

observations, and issue an alarm if the calculated probability of *not* detecting a change-point falls below a certain predefined, usually small, threshold (or equivalently, if this test statistic exceeds a certain threshold).

Change-point detection techniques have appeared useful in the networking context; we mention a few examples here. In [69] techniques from industrial statistics (such as control charts) have been proposed to detect anomalies in data collected in an ISP backbone network; see also [59] for a related study. A similar approach, in combination with queueing-theoretical results, has been developed (and extensively validated) for detecting overloads caused by VoIP traffic [65], [67]. We also mention [89], in which an anomaly-based intrusion detection systems is described, and tested in a SSH case study. At the methodological level, we also refer to the techniques proposed in [92].

The next Section, Section 4.1, introduces the problem and the evaluation criteria in terms of hypothesis testing. Section 4.2 discusses the celebrated cumulative sum (CUSUM) method. CUSUM is a *parametric technique* (a method that assumes an underlying distribution which is known to the observer) which can be used to detect changes in mean, variance, or the full distribution. Section 4.3 describes two *non-parametric techniques* (also known as distribution-free methods as no knowledge of underlying distributions is assumed) to detect a change in mean, while Section 4.4 concentrates on methods to detect a change in variance. The Section is concluded by a brief discussion of the AD methods described herein.

4.1 The Change-point Detection Problem

In this Section we give a mathematical formulation of the change-point detection problem. We look at the problem from a hypothesis testing point of view. In order to be able to assess the performance of the methods described in the following Sections, we also describe the evaluation criteria (the detection speed and the false alarm ratio) in mathematical terms.

Problem Description. We consider a sequence of observations $\mathbf{X} = (X_0, X_1, \dots)$, during which potentially a change-point occurs. In probabilistic terms such a change-point, to be considered as a change in the statistical law of the underlying random variable, can be described as follows.

Definition 1 [*Change-point*] Suppose there is a $k > 0$ such that the X_i are independent³ and identically distributed (i.i.d.) realizations of a random variable with density $f(\cdot)$ for $i = 0, \dots, k - 1$, while X_i are i.i.d. with a different density $g(\cdot)$ for $i \geq k$. In this case we call k a *change-point*.

At a point in time $n = 1, 2, \dots$ we check whether a change-point has occurred at some time $k \leq n$, by evaluating $\mathbf{X}_n = (X_0, \dots, X_n)$; if not we continue by evaluating $\mathbf{X}_{n+1} = (X_0, \dots, X_{n+1})$, etc. In terms of hypothesis testing: at time n we want to decide between two hypotheses:

³ There are also known results for a change-point detection in dependent data, see for example [8, Ch. II].

- Under the null-hypothesis (H_0) the X_i ($i = 0, \dots, n$) are i.i.d. realizations of a random variable with density $f(\cdot)$.
- Under the alternative hypothesis (H_1) there is a $1 \leq k \leq n$ such that up to $k - 1$ the observations are i.i.d. samples from a distribution with density $f(\cdot)$, while from observation k on they are i.i.d. with a *different* density $g(\cdot)$.

In other words: under the null-hypothesis there has *not* been a changepoint, while under the alternative hypothesis the process changes at some time k . This setup is not a simple binary hypothesis testing problem, but a *multiple-hypothesis test*, as the alternative is essentially a *union* of hypotheses. More precisely: if $H_1(k)$ corresponds to having a changepoint at k , we can write H_1 as the union of the $H_1(k)$, with $k = 1, \dots, n$.

Evaluating Changepoint Detection Methods. In statistics it is common to evaluate a hypothesis test by considering the probability of so-called type I and type II errors. As we perform a hypothesis test at each time step n (as long as no changepoint has been detected), the performance of the test at time n can be quantified by these error probabilities. They are defined as follows.

Definition 2 [*Type I and II errors*]

- A *type I error* (or: *false positive*) occurs if we decide to reject H_0 while it is actually true, in other words, if we detect a changepoint that has not occurred. The *type I error probability* is the probability that we reject H_0 under H_0 .
- A *type II error* (or: *false negative*) occurs if we decide to accept H_0 while it is not true, in other words, if we miss a changepoint. The *type II error probability* is the probability that we accept H_0 under H_1 .

Now consider a changepoint detection procedure for the sequence \mathbf{X} as a whole, that is, a stopping rule that issues an alarm at time $\tau \geq 1$, defined as the first time that we decide to reject H_0 . On the one hand τ should occur soon after the changepoint k , on the other hand, the rate of false alarms should be low. Mathematically, this can be formulated as keeping the distribution of $\tau - k$ stochastically small, given that the changepoint takes place at k (i.e., under $H_1(k)$), whereas the distribution of τ should be stochastically large in case there is *no* changepoint (i.e., under H_0). The criterion to manage the trade-off between detecting a changepoint early and to control the number of false alarms is translated in more formal terms as follows:

- minimize $\sup_{k \geq 1} \mathbb{E}_k(\tau - k + 1 \mid \tau \geq k)$ (meaning that time of detection is as soon as possible after the changepoint),
- while at the same time making sure that $\mathbb{E}_0\tau$ tends to be large (meaning that in case there is no changepoint, there is a strong tendency to not issue an alarm).

In this Section \mathbb{E}_k and \mathbb{P}_k stand for expectation and probability, respectively, under H_0 for $k = 0$ and under $H_1(k)$ for $k \geq 1$.

4.2 A Parametric Method: CUSUM

The method described in this Section, known as *Cumulative Sum* (or briefly CUSUM), has been proposed to identify a change in distribution. The method assumes that the

densities $f(\cdot)$ and $g(\cdot)$ are known. First we introduce the method, roughly following the setup presented in [88, Ch. II.6]. Then two approaches to approximate the detection speed and the false alarm rate are considered. For various examples of this approach, we refer to [65] and [67].

The CUSUM-Method. Let us consider the common likelihood ratio test for H_0 versus $H_1(k)$ (with H_0 and $H_1(k)$ as defined in Section 4.1). This test raises an alarm if the likelihood ratio

$$\bar{S}_k := \frac{\mathbb{P}_k(\mathbf{X}_n)}{\mathbb{P}_0(\mathbf{X}_n)} = \frac{\prod_{i=0}^n \mathbb{P}_k(X_i)}{\prod_{i=0}^n \mathbb{P}_0(X_i)} = \prod_{i=k}^n \frac{g(X_i)}{f(X_i)}$$

exceeds a certain value $\bar{b} > 1$. It turns out, though, that it is more practical to work with the corresponding *log-likelihood*:

$$S_k := \log \bar{S}_k = \sum_{i=k}^n \log \left(\frac{g(X_i)}{f(X_i)} \right).$$

To deal with the fact that H_1 equals the union of the $H_1(k)$, we have to verify whether there is a $k \in \{1, \dots, n\}$ such that S_k exceeds a certain critical value $b > 0$. As a result, the statistic for the *composite* test (that is, H_0 versus H_1) is

$$t_n := \max_{k \in \{1, \dots, n\}} S_k.$$

Using this test statistic, one could decide to reject H_0 at time n if $t_n \geq b$, for some value $b > 0$ that needs to be selected (in a way that properly balances the false alarm ratio and detection speed). This leads to the following method for AD.

Method 1. For a given threshold $b > 0$ and a sequence $\mathbf{X} = (X_0, X_1, \dots)$ the *CUSUM-method* raises an alarm at epoch τ , defined as

$$\tau := \inf \left\{ n : \max_{k \in \{1, \dots, n\}} \sum_{i=k}^n \log \left(\frac{g(X_i)}{f(X_i)} \right) \geq b \right\}.$$

Evidently, no alarm is raised if the sequence $(\max_{k \in \{1, \dots, n\}} S_k)_{n \in \mathbb{N}}$ never exceeds b . The interpretation is that we issue an alarm if, popularly speaking, there is a point k such that it is more likely that the observations from k on originate from a distribution with density $g(\cdot)$, than from a distribution with density $f(\cdot)$.

The test statistic t_n can be rewritten in terms of the cumulative sums $T_k = \sum_{i=0}^k \log g(X_i)/f(X_i)$ (with increments that are distributed as $Y_i = \log g(X_i)/f(X_i)$) — which explains the name of the test — as follows

$$t_n = T_n - \min_{k \in \{1, \dots, n\}} T_{k-1}. \tag{9}$$

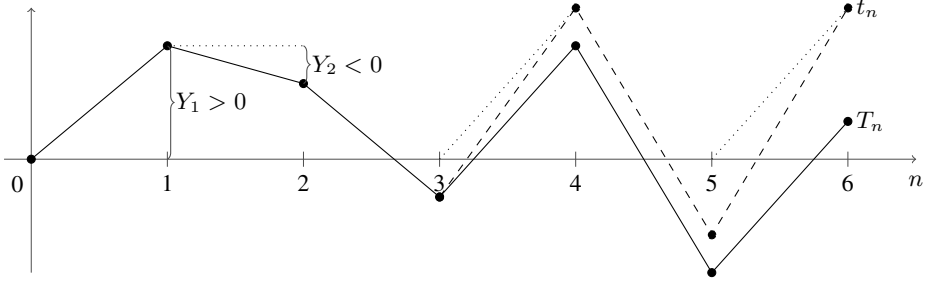


Fig. 5. An example realization of the processes t_n and T_n with regeneration points at $n = 3$ and $n = 5$

The statistic (9) represents the height of the random walk T_k (with $T_0 = 0$ and $T_i = T_{i-1} + Y_i$) with respect to the minimum that was achieved so far (among T_0, \dots, T_{k-1}); in this sense, there is a close connection to an associated (discrete-time) *queueing* process.

We continue by deriving expressions for $\mathbb{E}_0\tau$ (the expected time before an alarm is raised when no change occurs) and $\sup_{k \geq 1} \mathbb{E}_k(\tau - k + 1 \mid \tau \geq k)$ (the expected time between the changepoint and its detection). For this derivation it is important to notice that t_n has a *regeneration point* at j if $T_j = \min_{k \in \{1, \dots, j\}} T_k$. This means that the process t_n ‘forgets’ its past after time j , i.e. for $n > j$ it holds that t_n does not depend on the values Y_i for $i \leq j$, more precisely

$$t_n = \hat{T}_n - \min_{k \in \{j+1, \dots, n\}} \hat{T}_{k-1},$$

with $\hat{T}_k = \sum_{i=j+1}^k Y_i$. The concept of a regeneration point is illustrated in Fig. 5. We see that the process t_n is positive at time j or otherwise j is a minimum so far and thus a regeneration point (at which the process is ‘reset’).

We have that

$$\mathbb{E}_k(\tau - k + 1 \mid \tau \geq k) \leq \mathbb{E}_1\tau, \tag{10}$$

with equality if t_n has a regeneration point at $k - 1$. If $k - 1$ is not a regeneration point (t_{k-1} is positive), t_n (for $n \geq k$) becomes larger and τ smaller. It follows from (10) that

$$\sup_{k \geq 1} \mathbb{E}_k(\tau - k + 1 \mid \tau \geq k) = \mathbb{E}_1\tau.$$

Hence, to assess the performance of the test, we only have to analyze $\mathbb{E}_0\tau$ and $\mathbb{E}_1\tau$.

These quantities can be evaluated as follows. Let us start with analyzing $\mathbb{E}_0\tau$. Define $M_0 := 0$, and define (recursively) M_k as the first number n after M_{k-1} that t_n either becomes negative (meaning that a minimum is achieved), or exceeds b (meaning that an alarm is issued), and set $N_k := M_k - M_{k-1}$. Due to the underlying regenerative structure and the fact that all X_i are i.i.d., the N_k are i.i.d. as well (distributed as a random variable N); the number of N_k needed to exceed b is geometrically distributed

with success probability $\mathbb{P}_0(T_N \geq b)$. It is now an immediate consequence of Wald's equation that

$$\mathbb{E}_0\tau = \frac{\mathbb{E}_0N}{\mathbb{P}_0(T_N \geq b)};$$

and, analogously, $\mathbb{E}_1\tau = \mathbb{E}_1N/\mathbb{P}_1(T_N \geq b)$.

These expressions nicely illustrate the trade-off between timely detection and an increased rate of false alarms. As both expressions are increasing in b , the lower b , the faster the test reacts to a change (the smaller $\mathbb{E}_1\tau$), but the larger the number of false alarms (the smaller $\mathbb{E}_0\tau$). One could for instance pick the lowest b for which $\mathbb{E}_0\tau$ remains above a given (large) threshold B .

In [88, Ch. II.6] approximations for \mathbb{E}_0N , $\mathbb{P}_0(T_N \geq b)$, \mathbb{E}_1N and $\mathbb{P}_1(T_N \geq b)$ in terms of b , $f(\cdot)$ and $g(\cdot)$ are given by considering the sequential probability ratio test with a binary hypothesis (H_0 versus $H_1(1)$), test statistic T_n , lower bound 0 and upper bound b). In the following Sections we discuss other ways to approximate $\mathbb{E}_0\tau$ and $\mathbb{E}_1\tau$. Procedures in the spirit of CUSUM date back to at least [73]. The optimality of CUSUM (in terms of balancing fast detection with a low number of false alarms, as described above) was considered in a Bayesian framework by Shiryaev [84,85]. Lorden [61] (focusing on the limiting case that the threshold B tends to ∞) and Pollak [74] address the non-Bayesian setting.

Brownian Approximation of CUSUM. As $\mathbb{E}_0\tau$ and $\mathbb{E}_1\tau$ are in general hard to compute in explicit terms, one could opt for approximating them. In this Section we point out how to approximate them relying on a Brownian approximation (whose applicability is justified by the central limit theorem); in the next Section we turn to a large-deviations based approximation.

Define μ and σ^2 as the mean and variance (assumed to exist) of $\log g(X_i)/f(X_i)$ if the X_i are distributed according to density $f(\cdot)$, and $\bar{\mu}$ and $\bar{\sigma}^2$ if the X_i are distributed according to density $g(\cdot)$. Let, as before, τ denote the first time t_n exceeds b . To facilitate explicit computations, we now approximate, under \mathbb{P}_0 ,

$$T_k \stackrel{d}{\approx} \sigma B_k + \mu k,$$

with B_k standard Brownian motion; as mentioned earlier, the justification of this approximation lies in the central limit theorem (CLT). We thus obtain (the first equality being a standard identity)

$$\mathbb{E}_0\tau = \int_0^\infty \mathbb{P}_0(\tau > t) dt \tag{11}$$

$$\approx \int_0^\infty \mathbb{P}_0 \left(\max_{u \in [0,t]} \left((\sigma B_u + \mu u) - \min_{s \in [0,u]} (\sigma B_s + \mu s) \right) < b \right) dt. \tag{12}$$

It is a known fact (see e.g. [64, Ch. XII.1] or [88, Eqn. (3.15)]) that

$$\mathbb{P}_0 \left(\max_{s \in [0,t]} \sigma B_s + \mu s < b \right) = \Phi \left(\frac{bt - \mu}{\sigma\sqrt{t}} \right) - e^{2b\mu/\sigma^2} \Phi \left(\frac{bt + \mu}{\sigma\sqrt{t}} \right), \tag{13}$$

with $\Phi(\cdot)$ the cumulative distribution function of the standard Normal distribution; there is no straightforward way to evaluate (11) though. In various other asymptotic regimes the quantities $\mathbb{E}_0\tau = \mathbb{E}_0N/\mathbb{P}_0(T_N \geq b)$ and $\mathbb{E}_1\tau = \mathbb{E}_1N/\mathbb{P}_1(T_N \geq b)$ can be analyzed; see e.g. [88, Ch. X.2].

It is possible to accurately approximate $\mathbb{P}_0(t_n \geq b)$, using the above CLT-based methodology. Due to reversibility arguments,

$$\begin{aligned} t_n &= T_n - \min_{k \in \{1, \dots, n\}} T_{k-1} = \max_{k \in \{1, \dots, n\}} (T_n - T_{k-1}) \\ &=_{\text{d}} \max_{k \in \{1, \dots, n\}} T_{n-k+1} = \max_{k \in \{1, \dots, n\}} T_k, \end{aligned}$$

so that

$$\mathbb{P}_0(t_n \geq b) = \mathbb{P}_0(\exists k \in \{1, \dots, n\} : T_k \geq b) \approx \mathbb{P}_0(\exists s \in [0, n] : \sigma B_s + \mu s \geq b);$$

then Eqn. (13) can be applied immediately. Similar arguments can be used to assess $\mathbb{P}_0(t_n \geq b(n))$ if an alarm is issued as soon as t_n exceeds the straight line $b(n) = b_0 + b_1n$. We then have to replace μ in the above setup by $\mu - b_1$, while we set $b := b_0$. One could even consider curved boundaries, cf. [100].

Large-Deviations Approximation of CUSUM. Above we pointed out how to approximate the probability $\mathbb{P}_0(t_n \geq b)$ in a central-limit regime, relying on a Brownian approximation. This Section presents an alternative approach, based on large-deviations asymptotics [10,34,64]. A key step is that we scale b by n , and focus on asymptotics of the probability of issuing a false alarm at time n , that is, $\mathbb{P}_0(t_n \geq nb)$ for large n ; we roughly follow the setup of [10, Ch. VI.E]. As we saw before, this probability can be rewritten as

$$\mathbb{P}_0(t_n \geq nb) = \mathbb{P}_0(\exists k \in \{1, \dots, n\} : T_k \geq nb).$$

Due to $n^{-1} \cdot \log n \rightarrow 0$ and

$$\max_{k \in \{1, \dots, n\}} \mathbb{P}_0(T_k \geq nb) \leq \mathbb{P}_0(\exists k \in \{1, \dots, n\} : T_k \geq nb) \leq n \cdot \max_{k \in \{1, \dots, n\}} \mathbb{P}_0(T_k \geq nb),$$

we have that

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}_0(t_n \geq nb) = \max_{\lambda \in [0,1]} \lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}_0\left(\frac{T_{n\lambda}}{n\lambda} \geq \frac{b}{\lambda}\right)$$

(realize that $n\lambda$ is not necessarily integer, so there is mild abuse of notation in the previous display). Relying on Cramér’s theorem, we can rewrite this to

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}_0(t_n \geq nb) &= \max_{\lambda \in [0,1]} \lim_{n \rightarrow \infty} \frac{\lambda}{n\lambda} \log \mathbb{P}_0\left(\frac{T_{n\lambda}}{n} \geq b\right) \\ &= \max_{\lambda \in [0,1]} \left(-\lambda \sup_{\theta} \left(\theta \frac{b}{\lambda} - \log M(\theta)\right)\right); \end{aligned}$$

here $M(\theta)$ is the moment generating function (under H_0) of $\log g(X_i)/f(X_i)$:

$$M(\theta) = \mathbb{E}_0 \exp\left(\theta \log \frac{g(X_i)}{f(X_i)}\right) = \mathbb{E}_0 \left(\frac{g(X_i)}{f(X_i)}\right)^\theta = \int_{-\infty}^{\infty} (g(x))^\theta (f(x))^{1-\theta} dx.$$

We can then set b such that the decay rate under study equals some predefined (negative) constant $-\alpha$. In principle, however, there is no need to take a constant b ; we could pick a function $b(\cdot)$ instead. It can be seen that, in terms of optimizing the type II error performance (i.e. not detecting a changepoint), it is optimal to choose the function $b(\cdot)$ such that

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}_0 \left(\frac{T_{n\lambda}}{n} \geq b(\lambda) \right) = -\lambda \sup_{\theta} \left(\theta \frac{b(\lambda)}{\lambda} - \log M(\theta) \right) \tag{14}$$

is *constant* in $\lambda \in [0, 1]$ (and equaling $-\alpha$); intuitively, this entails that at any point $n\lambda$ in time, issuing an alarm (which is done if $T_n - T_{n\lambda}$ exceeds $nb(1 - \lambda)$) is essentially equally likely.

The above mentioned strong type II error performance can also be understood as follows. Suppose there is a changepoint at $k = n\lambda^*$. Then the mean of $T_n - T_{n\lambda^*}$ equals

$$n(1 - \lambda^*) \int_{-\infty}^{\infty} \log \left(\frac{g(x)}{f(x)} \right) g(x) dx.$$

The expression in the previous display equals $n(1 - \lambda^*)M'(1)$. On the other hand, the optimizing $\theta \equiv \theta_\lambda$ in (14) is such that

$$\frac{b(\lambda)}{\lambda} = \frac{M'(\theta_\lambda)}{M(\theta_\lambda)}.$$

Due to the fact that this optimizing θ lies between 0 and 1, and that $M'(\theta)/M(\theta)$ is increasing in θ (use that the moment generating function $M(\theta)$ is convex!), we obtain that

$$b(\lambda) = \lambda \frac{M'(\theta_\lambda)}{M(\theta_\lambda)} \leq \lambda \frac{M'(1)}{M(1)} = \lambda M'(1).$$

Combining the above, we conclude that the mean of $T_n - T_{n\lambda^*}$, given that there is a changepoint at $k = n\lambda^*$, is larger than $nb(1 - \lambda^*)$. As a consequence, with high probability an alarm is issued.

As an illustration, let us identify the function $b(\cdot)$ in a specific example. Let $f(\cdot)$ correspond with the standard Normal density, while $g(\cdot)$ corresponds with a Normal density with mean μ and unit variance. It is seen that

$$M(\theta) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp \left(-\frac{\theta}{2}(x - \mu)^2 \right) \exp \left(-\frac{1-\theta}{2}x^2 \right) = \exp \left(-\frac{1}{2}\theta(1 - \theta)\mu^2 \right).$$

As a consequence, we need to derive b from

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}_0 \left(\frac{T_{n\lambda}}{n} \geq b(\lambda) \right) = \theta_b b - \frac{\lambda}{2} \theta_b (1 - \theta_b) \mu^2 = \alpha,$$

with $\theta_b = b/(\lambda\mu^2) + \frac{1}{2}$, which amounts to solving a trivial quadratic equation.

4.3 Non-parametric Methods for Detecting Changes in Mean

The CUSUM technique described in the previous Section can be considered as an example of *parametric* statistics: it is assumed that the data stem from a type of probability distribution which is known in advance. This Section is about *non-parametric* statistics, in which there are no *a priori* assumptions on data belonging to a particular distribution. Two non-parametric methods are described in this Section, both can be used to detect changes in the mean value.

The Method of Brodsky/Darkhovsky. The first non-parametric method to discuss is an algorithm studied in [8, Ch. IV.1]. At time n their method considers the observations $X_{n-N+1}, X_{n-N+2}, \dots, X_n$, where N is the so-called window size. In order to check whether a changepoint has occurred at time $n - N + k + 1$, the average over $X_{n-N+1}, \dots, X_{n-N+k}$ is compared with the average over the rest of the values in the window $X_{n-N+k+1}, \dots, X_n$. If there is no changepoint, the difference tends to be close to 0. Therefore an alarm is raised if there is a k for which the difference exceeds some threshold $c > 0$. For k close to 1 or $N - 1$ one of the averages contains few values therefore a parameter $\gamma \in (0, \frac{1}{2})$ is chosen and only values $k \in \{\lceil \gamma N \rceil, \dots, \lfloor (1 - \gamma)N \rfloor\}$ are considered. The method, illustrated in Fig. 6, is summarized as follows.

Method 2. Fix the window width N , $\gamma \in (0, \frac{1}{2})$ and threshold $c > 0$. Introduce

$$Y_n(k, N) := \frac{1}{k} \sum_{i=n-N+1}^{n-N+k} X_i - \frac{1}{N-k} \sum_{i=n-N+k+1}^n X_i.$$

The *method of Brodsky/Darkhovsky* prescribes to issue an alarm if the test statistic

$$Y_n(N) := \max_{k \in \{\lceil \gamma N \rceil, \dots, \lfloor (1-\gamma)N \rfloor\}} |Y_n(k, N)|$$

exceeds the threshold c .

Suppose that before the changepoint the mean is (without loss of generality) 0, and after the changepoint $\mu > 0$. It is required that $\mu > c$, as otherwise issuing an alarm remains rare, even if there *is* a changepoint. An alarm is issued about cN/μ timeslots after the changepoint.

To assess the type I performance of the test, we now analyze $\mathbb{P}_0(Y_n(N) \geq c)$. A large-deviations based evaluation of this probability is straightforward in case it is assumed that the moment generating function is finite around the origin (that is, $M(\theta) := \mathbb{E}_0 e^{\theta X_i} < \infty$ for some positive θ). To this end, first observe that $\pi(N) \leq \mathbb{P}_0(Y_n(N) \geq c) \leq N(1 - 2\gamma)\pi(N)$, where (with mild abuse of notation)

$$\pi(N) := \max_{f \in (\gamma, 1-\gamma)} \mathbb{P}_0 \left(\frac{1}{f} \sum_{i=1}^{fN} X_i + \frac{1}{1-f} \sum_{i=fN+1}^N X_i \geq cN \right);$$

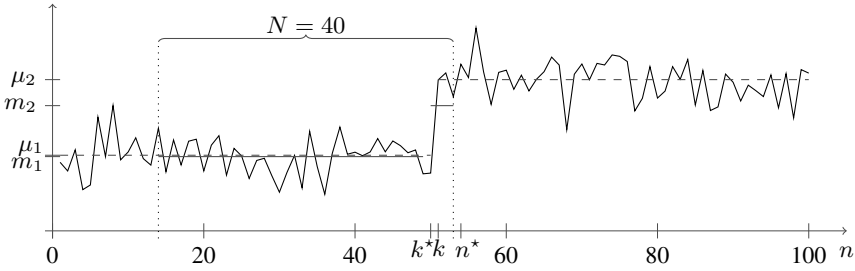


Fig. 6. An example where the method of Brodsky/Darkhovsky is used to detect a change in mean from $\mu_1 = 1$ to $\mu_2 = 2$ at time $k = 51$. The settings are $N = 50$, $\gamma = 0.1$ and $c = 0.5$. A changepoint at $k^* = 50$ is detected at time $n^* = 53$. The difference between the sample averages $m_1 = \frac{1}{46} \sum_{i=14}^{49} X_i$ and $m_2 = \frac{1}{4} \sum_{i=50}^{53} X_i$ is $1.65 - 0.98 = 0.67$ which is larger than the threshold.

$\mathbb{P}_0(Y_n(N) \leq -c)$ can be dealt with similarly. Because of Cramér’s theorem,

$$\begin{aligned} & \lim_{N \rightarrow \infty} \frac{1}{N} \log \mathbb{P}_0 \left(\frac{1}{f} \sum_{i=1}^{fN} X_i + \frac{1}{1-f} \sum_{i=fN+1}^N X_i \geq cN \right) \\ &= - \sup_{\theta} \left(\theta c - f \log M \left(\frac{\theta}{f} \right) - (1-f) \log M \left(\frac{\theta}{1-f} \right) \right). \end{aligned}$$

Under mild conditions, the optimum over f is achieved at $f = \frac{1}{2}$, so that we arrive at

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log \mathbb{P}_0(Y_n(N) \geq c) = - \sup_{\theta} (\theta c - \log M(2\theta)).$$

A Non-parametric Version of CUSUM. A non-parametric alternative to the CUSUM technique that we described in Section 4.2, is to replace the increments $Y_i = \log f(X_i)/g(X_i)$ by the X_i themselves, see e.g. [8, Ch. IV.2].

Method 3. Non-parametric CUSUM issues an alarm as soon as

$$U_n := \max_{k \in \{1, \dots, n\}} \sum_{i=k}^n X_i$$

exceeds a given threshold c .

The test statistic U_n can be written in terms of the cumulative sums $\bar{T}_k := \sum_{i=0}^{k-1} X_i$:

$$U_n = \bar{T}_{n+1} - \min_{k \in \{1, \dots, n\}} \bar{T}_k.$$

The test is used to detect a change in mean from a distribution with $\mu \leq 0$ to a distribution with $\mu > 0$. The idea behind the method is that the expectation of $\sum_{i=k}^n X_i$ is zero or less for all $k \in \{1, \dots, n\}$ if no changepoint has occurred, while the expectation of $\sum_{i=k}^n X_i$ grows with n if at time k a changepoint has taken place; therefore $\sum_{i=k}^n X_i$ must cross c for some time n . An advantage of non-parametric CUSUM is the fact that it does not require the X_i to be independent; instead it is sufficient if a specific mixing condition applies.

If there is a changepoint at epoch k to a distribution with mean $\mu > 0$, then this is detected around c/μ units later. Various queueing-theoretic results can be used to further sharpen these results.

To determine the type I error, assume without loss of generality that $\mathbb{E}_0 X_i < 0$. In case the X_i have a finite moment generating function (again, $M(\theta) := \mathbb{E}_0 e^{\theta X_i} < \infty$ for some positive θ), then the type I error can be assessed through (using the Markov inequality)

$$\begin{aligned} \mathbb{P}_0(\exists k \in \{1, \dots, n\} : \bar{T}_k \geq c) &\leq \sum_{k=1}^n \mathbb{P}_0(\bar{T}_k \geq c) \\ &\leq \sum_{k=1}^n \frac{e^{-\theta kc}}{(M(\theta))^k} = ne^{-\theta c}, \end{aligned}$$

so that we have the following upper bound on the decay rate of the probability of a false alarm under H_0 , cf. [8, Thm. 4.2.1]:

$$\lim_{c \rightarrow \infty} \frac{1}{c} \log \mathbb{P}_0(U_n \geq c) \leq -\max\{\theta : M(\theta) < \infty\}.$$

4.4 Methods for Detecting Changes in Variance

In order to identify changes in the variance (with the mean fixed), various approaches have been proposed. Assuming without loss of generality that the mean is 0, the general idea is that we keep track of the *sum of squares* $V_k := \sum_{i=1}^k X_i^2$. One idea would be to apply CUSUM to this sequence, but various other tests have been proposed. All the methods presented below assume a Normal distribution of the observed data.

Hsu [47] proposes to consider the statistic

$$v_k := \frac{V_n - V_k}{V_k} \frac{k}{n - k};$$

its distribution under H_0 (that is, no change in variance) can be expressed in terms of an F -distribution. The resulting test is most powerful in terms of detecting a (one-sided) shift at a prescribed position k . Hsu [47] then points out how to adapt these statistics v_k into a single statistic v to be used for detecting a variance changepoint somewhere in the sequence X_1, \dots, X_n (where obtaining explicit distributional properties under H_0 of the resulting test statistic is somewhat problematic).

The procedure proposed by Inclán and Tiao [49] uses a similar test statistic, for which the (approximate) distribution under H_0 can be derived. They define

$$\bar{v}_k := \frac{V_k}{V_n} - \frac{k}{n};$$

notice that this statistic has value 0 at $k = 0$ and $k = n$. It is shown in [49] that \bar{v}_{tn} (with $t \in [0, 1]$) converges in distribution to a *Brownian Bridge*, that is, a Brownian motion $(B_t)_t$ (with $\text{Var}B_t = \sigma^2 t$) conditioned on $B_1 = 0$. This means that we have an adequate approximation for the test statistic $\bar{v} := \max_{k \in \{1, \dots, n\}} \bar{v}_k$, as

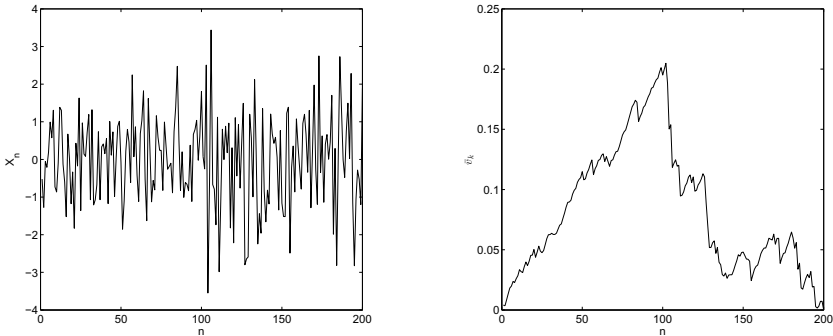
$$\mathbb{P} \left(\max_{t \in [0,1]} B_t \geq x \mid B_1 = 0 \right) = e^{-2x^2/\sigma^2};$$

for the distribution of $\max_{t \in [0,1]} |B_t|$ see [78, Ch. VI.10]. An information-criterion based technique has been proposed by Chen and Gupta [23]. There the test statistic essentially reads

$$\check{v}_k := k \log \frac{V_k}{k} + (n - k) \log \frac{V_n - V_k}{n - k}.$$

Distributional properties of $\check{v} := \max_{k \in \{1, \dots, n\}} \check{v}_k$ under H_0 are derived ($n \rightarrow \infty$), in terms of a specific Gumbel distribution. The procedure can be used to detect multiple changepoints as well.

Fig. 7 shows a sample dataset with the first 100 observations distributed according to $\mathbb{N}(0, 1)$ and next 100 according to $\mathbb{N}(0, 2)$, along with the test statistics \bar{v}_k by Inclán and Tiao. The point at which a maximum of \bar{v}_k is attained ($n = 102$) is a reported changepoint position. For this particular dataset, Chen and Gupta method also returned $n = 102$ as a result.



(a) Variance change at $n = 101$

(b) Test statistics with changepoint detected at $n = 102$

Fig. 7. Gaussian data with changepoint in variance

4.5 Discussion

In this Section we have presented several statistical methods for the detection of sudden and insistent changes (changepoints) in sequences of observations. The methods differ in the type of changes that can be detected (e.g. change in distribution, mean, or variance) and the requirements on the input (e.g. independent observations, known

distribution). We conclude this Section with a qualitative comparison between the presented methods. In addition to the two above-mentioned points (type of change and input requirements) we also discuss for each method what parameters it has and what kind of results are available regarding its performance in terms of detection speed and false alarm ratio.

CUSUM [88] is designed to detect a change in distribution. As a change in mean or variance changes the distribution, the method can also be used to detect these types of changes. The method has two strong assumptions. Firstly, it assumes that the observations are independent and, secondly, it assumes that the distributions before and after the change are known (or estimated). Therefore CUSUM may perform worse if there is a change to a distribution different than the assumed distribution. The method includes a threshold on the test statistic, which is (informally speaking) a threshold on the ratio between the maximum (over all points in time) of the probability that there is a changepoint at that specific time and the probability that there is no changepoint in the data. When setting the threshold, it should be taken into account that the value of this parameter affects both the detection speed and the number of false alarms. We have discussed two ways to approximate these quantities as a function of the threshold.

The method of Brodsky & Darkhovsky [8] aims at finding a change in the mean of a time series. For this method, no distributions need to be estimated. The method requires the user to set three parameters. Firstly, one has to decide on the window size and, secondly, on a parameter γ . A window of observations is divided into two intervals such that both of the contain at least a fraction γ of the number of observations in the window. Thirdly, a threshold on the change in average needs to be chosen; this threshold determines when an alarm is raised. We have given an expression for the expected detection speed and an approximation of the probability of a false alarm; these can be used to choose the threshold; in the corresponding derivations independence is assumed.

A second distribution-free method to detect a change in mean is based on the CUSUM method. This method [8, Ch. IV.2] — referred to as non-parametric CUSUM — has one parameter only, viz. the threshold imposed on the test statistic. An alarm is raised if the difference between the cumulative sum of the observed values and the minimum so far exceeds this threshold. Also for this method we have given an expression for the expected detection speed and an approximation of the probability of a false alarm as functions of the threshold. In this case, the expressions do not assume independence; it is sufficient if a specific mixing condition applies.

We have presented three methods for detecting change in variance: the methods of Hsu [47], Inclán & Tiao [49], and Chen & Gupta [23]. All of these methods assume the observed data to be normally distributed (see, however, the above cited papers on some discussion on possibility of relaxing this restriction) and rely on a test statistic — for a changepoint at a certain time k — which is a function of the sum of the squares of the observed values up to this time k and the sum of squares up to the current observation. An alarm is raised if there is a time k for which the test statistic exceeds the threshold. This threshold is the only parameter of the methods. For all methods the probability of a false alarm can be approximated either for the *single test* finding changepoints at a *predefined time k* or the *multiple test* finding changepoint at *some time k* .

5 Conclusions

In this Chapter we have discussed different ways in which the problem of anomaly detection can be faced, also taking into account the constraints imposed by the different network scenarios (i.e., backbone and access networks).

Roughly speaking, the Chapter has addressed three different aspects, which rarely are covered together:

- broad overview of the statistical methods applied to AD;
- discussion of the network operator perspective, covering both operational issues and practical guidelines for designing an AD tool;
- deep analysis, from a mathematical point of view, of one of the most promising AD methods.

Ideally, the aim of this Chapter is to allow a reader not only to understand the described methods, but also to help her to grasp the complexity behind the design and the application of an AD method.

References

1. Darpa intrusion detection evaluation data set, <http://www.ll.mit.edu/mission/communications/ist/corpora/ideval>
2. Kdd cup (1999), data, <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
3. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: SODA 2007: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035. Society for Industrial and Applied Mathematics, Philadelphia (2007)
4. Barford, P., Kline, J., Plonka, D., Ron, A.: A signal analysis of network traffic anomalies. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, IMW 2002, pp. 71–82. ACM, New York (2002)
5. Borgnat, P., Dewaele, G., Fukuda, K., Abry, P., Cho, K.: Seven years and one day: Sketching the evolution of internet traffic. In: INFOCOM (April 2009)
6. Bouzida, Y., Cuppens, F., Cuppens-Bouahia, N.A., Gombault, S.N.: Efficient intrusion detection using principal component analysis. In: 3ème Conférence sur la Sécurité et Architectures Réseaux, La Londe, France, Juin, RSM - Dépt. Réseaux, Sécurité et Multimédia (Institut Télécom-Télécom Bretagne) (2004)
7. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. ACM SIGMOD Record 29(2), 93–104 (2000)
8. Brodsky, B., Darkhovsky, B.: Nonparametric Methods in Change-point Problems. Kluwer (1993)
9. Brown, C., Cowperthwaite, A., Hijazi, A., Somayaji, A.: Analysis of the 1999 darpa/lincoln laboratory ids evaluation data with netadict. In: CISDA 2009: Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications, pp. 67–73. IEEE Press, Piscataway (2009)
10. Bucklew, J.: Large Deviation Techniques in Decision, Simulation, and Estimation. Wiley (1985)
11. Burgess, M., Haugerud, H., Straumsnes, S., Reitan, T.: Measuring system normality. ACM Trans. Comput. Syst. 20(2), 125–160 (2002)

12. Callegari, C., Gazzarrini, L., Giordano, S., Pagano, M., Pepe, T.: A novel multi time-scales pca-based anomaly detection system. In: 2010 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS (2010)
13. Callegari, C., Gazzarrini, L., Giordano, S., Pagano, M., Pepe, T.: When randomness improves the anomaly detection performance. In: Proceedings of 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies, ISABEL (2010)
14. Callegari, C., Giordano, S., Pagano, M.: Application of Wavelet Packet Transform to Network Anomaly Detection. In: Balandin, S., Moltchanov, D., Koucheryavy, Y. (eds.) NEW2AN 2008. LNCS, vol. 5174, pp. 246–257. Springer, Heidelberg (2008)
15. Callegari, C., Giordano, S., Pagano, M., Pepe, T.: On the use of sketches and wavelet analysis for network anomaly detection. In: IWCMC 2010: Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, pp. 331–335. ACM, New York (2010)
16. Callegari, C., Giordano, S., Pagano, M., Pepe, T.: Combining sketches and wavelet analysis for multi time-scale network anomaly detection. *Computers & Security* 30(8), 692–704 (2011)
17. Callegari, C., Giordano, S., Pagano, M., Pepe, T.: Detecting heavy change in the heavy hitter distribution of network traffic. In: IWCMC, pp. 1298–1303. IEEE Press (2011)
18. Callegari, C., Giordano, S., Pagano, M., Pepe, T.: Detecting anomalies in backbone network traffic: a performance comparison among several change detection methods. *IJNet* 11(4), 205–214 (2012)
19. Carl, G., Brooks, R.R., Rai, S.: Wavelet based denial-of-service detection. *Computers & Security* 25(8), 600–615 (2006)
20. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. *ACM Comput. Surv.* 41(3), 15:1–15:58 (2009)
21. Charikar, M., Chen, K., Farach-Colton, M.: Finding frequent items in data streams. In: Proc. VLDB Endow, pp. 693–703 (2002)
22. Chatzigiannakis, V., Papavassiliou, S., Androulidakis, G.: Improving network anomaly detection effectiveness via an integrated multi-metric-multi-link (m^3l) pca-based approach. *Security and Communication Networks* 2(3), 289–304 (2009)
23. Chen, J., Gupta, A.: Testing and locating variance change points with application to stock prices. *J. Am. Statist. Assoc.* 92, 739–747 (1997)
24. Cheung-Mon-Chan, P., Clerot, F.: Finding hierarchical heavy hitters with the count min sketch. In: Proceedings of 4th International Workshop on Internet Performance, Simulation, Monitoring and Measurement, IPS-MOME (2006)
25. Coifman, R.R., Wickerhauser, M.V.: Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory* 38(2), 713–718 (1992)
26. Cormode, G., Muthukrishnan, S.: What’s hot and what’s not: Tracking most frequent items dynamically. In: Proceedings of ACM Principles of Database Systems, pp. 296–306 (2003)
27. Cormode, G., Muthukrishnan, S.: What’s new: Finding significant differences in network data streams. In: Proc. of IEEE Infocom, pp. 1534–1545 (2004)
28. Cormode, G., Muthukrishnan, S.: An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms* 55(1), 58–75 (2005)
29. Cormode, G., Muthukrishnan, S., Srivastava, D.: Finding hierarchical heavy hitters in data streams. In: Proc. of VLDB, pp. 464–475 (2003)
30. Dainotti, A., Pescape, A., Ventre, G.: Wavelet-based detection of dos attacks. In: Proceedings of Global Telecommunications Conference, GLOBECOM 2006, pp. 1–6. IEEE (2006)
31. D’Alconzo, A., Coluccia, A., Romirer-Maierhofer, P.: Distribution-based anomaly detection in 3g mobile networks: from theory to practice. *Int. J. Netw. Manag.* 20(5), 245–269 (2010)

32. Daubechies, I.: Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics* 41, 909–996 (1988)
33. Daubechies, I.: *Ten lectures on Wavelets*. CBMS-NSF Series in Applied Mathematics, vol. 61. SIAM, Philadelphia (1992)
34. Dembo, A., Zeitouni, O.: *Large Deviations Techniques and Applications*. Springer (1998)
35. Denning, D.E.: An intrusion-detection model. *IEEE Transactions on Software Engineering* 13(2), 222–232 (1987)
36. Dewaele, G., Fukuda, K., Borgnat, P., Abry, P., Cho, K.: Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures. In: *LSAD 2007: Proceedings of the 2007 Workshop on Large Scale Attack Defense*, pp. 145–152. ACM, New York (2007)
37. Ensafi, R., Dehghanzadeh, S., Akbarzadeh, T.M.R.: Optimizing fuzzy k-means for network anomaly detection using pso. In: *AICCSA 2008: Proceedings of the 2008 IEEE/ACS International Conference on Computer Systems and Applications*, pp. 686–693. IEEE Computer Society, Washington, DC (2008)
38. Ertöz, L., Eilertson, E., Lazarevic, A., Tan, P.N., Kumar, V., Srivastava, J.P., Dokas, P.: *MINDS - Minnesota Intrusion Detection System*. MIT Press (2004)
39. Eskin, E., Arnold, A., Prerai, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In: *Applications of Data Mining in Computer Security*. Kluwer (2002)
40. Estan, C., Varghese, G.: New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Transactions on Computer Systems* 21, 270–313 (2003)
41. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise, pp. 226–231. AAAI Press (1996)
42. Fox, K.L., Henning, R.R., Reed, J.H., Simonian, R.P.: A neural network approach towards intrusion detection. In: *Proc. 13th National Computer Security Conference. Information Systems Security. Standards - the Key to the Future*, vol. I, pp. 124–134 (1990)
43. Maier, G., Feldmann, A., Paxson, V., Allman, M.: On dominant characteristics of residential broadband internet traffic. In: *IEEE IMC (2009)*
44. Gao, J., Hu, G., Yao, X.: Anomaly detection of network traffic based on wavelet packet (2006)
45. Gilbert, A.C.: Multiscale analysis and data networks. *Applied and Computational Harmonic Analysis* 10, 185–202 (2001)
46. Hodge, V., Austin, J.: A survey of outlier detection methodologies. *Artif. Intell. Rev.* 22(2), 85–126 (2004)
47. Hsu, D.: Tests for variance shift at an unknown time point. *Appl. Statist.* 26, 279–284 (1977)
48. Huang, P., Feldmann, A., Willinger, W.: A non-intrusive, wavelet-based approach to detecting network performance problems. In: *IMW 2001: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pp. 213–227 (2001)
49. Inclán, C., Tiao, G.: Use of cumulative sums of squares for retrospective detection of changes of variance. *J. Am. Statist. Assoc.* 89, 913–923 (1994)
50. Zaki, M.J., Sequeira, K.: Admit: Anomaly-base data mining for intrusions. In: *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Jul. 2002)*
51. Karp, R.M., Papadimitriou, C.H., Shenker, S.: A simple algorithm for finding frequent elements in streams and bags. *ACM Transactions on Database Systems* 28 (2003)
52. Kim, S.S., Narasimha Reddy, A.L., Vannucci, M.: Detecting traffic anomalies using discrete wavelet transform. In: *Proceedings of International Conference on Information Networking (ICOIN), Busan, Korea*, pp. 1375–1384 (2003)
53. Lakhina, A.: Diagnosing network-wide traffic anomalies. In: *ACM SIGCOMM*, pp. 219–230 (2004)

54. Lakhina, A., Crovella, M., Diot, C.: Characterization of network-wide anomalies in traffic flows. In: ACM Internet Measurement Conference, pp. 201–206 (2004)
55. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. In: ACM SIGCOMM (2005)
56. Lakhina, A., Papagiannaki, K., Crovella, M., Christophe, D., Kolaczyk, E.D., Taft, N.: Structural analysis of network traffic flows. In: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2004/Performance 2004, pp. 61–72. ACM, New York (2004)
57. Lazarevic, A., Ozgur, A., Ertoz, L., Srivastava, J., Kumar, V.: A comparative study of anomaly detection schemes in network intrusion detection. In: Proceedings of the Third SIAM International Conference on Data Mining (2003)
58. Leland, W.E., Taquq, M.S., Willinger, W., Wilson, D.V.: On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.* 2(1), 1–15 (1994)
59. Lin, S.-Y., Liu, J.-C., Zhao, W.: Adaptive cusum for anomaly detection and its application to detect shared congestion. Texas A&M University. Technical Report TAMU-CS-TR-2007-1-2 (2007)
60. Liu, Y., Zhang, L., Guan, Y.: Sketch-based streaming pca algorithm for network-wide traffic anomaly detection. In: Proceedings of International Conference on Distributed Computing Systems (2010)
61. Lorden, G.: Procedures for reacting to a change in distribution. *Ann. Math. Statist.* 42, 1897–1908 (1971)
62. Lu, W., Ghorbani, A.: Network anomaly detection based on wavelet analysis. *EURASIP Journal on Advances in Signal Processing* (1), 837601 (2009)
63. Mallat, S.G.: A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(7), 674–693 (1989)
64. Mandjes, M.: *Large Deviations for Gaussian Queues*. Wiley (2007)
65. Mandjes, M., Zuraniewski, P.: $M/g/\infty$ transience, and its applications to overload detection. *Performance Evaluation* 68, 507–527 (2011)
66. Manku, G.S., Motwani, R.: Approximate frequency counts over data streams. In: VLDB, pp. 346–357 (2002)
67. Mata, F., Zuraniewski, P., Mandjes, M., Mellia, M.: Anomaly detection in voip traffic with trends. In: Proceedings of the 24th International Teletraffic Congress (2012)
68. Matteoli, S., Diani, M., Corsini, G.: A tutorial overview of anomaly detection in hyperspectral images. *IEEE Aerospace and Electronic Systems Magazine* 25(7), 5–28 (2010)
69. Münz, G., Carle, G.: Application of forecasting techniques and control charts for traffic anomaly detection. In: Proceedings of the 19th ITC Specialist Seminar on Network Usage and Traffic (2008)
70. Munz, G., Li, S., Carle, G.: Traffic anomaly detection using k-means clustering. In: GI/ITG-Workshop MMBnet (2007)
71. Muthukrishnan, S.: Data streams: algorithms and applications. In: Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 413–413. Society for Industrial and Applied Mathematics, Philadelphia (2003)
72. Oldmeadow, J., Ravinutala, S., Leckie, C.: Adaptive Clustering for Network Intrusion Detection. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 255–259. Springer, Heidelberg (2004)
73. Page, E.: Continuous inspection scheme. *Biometrika* 41, 100–115 (1954)
74. Pollak, M.: Optimal detection of a change in distribution. *Ann. Statist.* 13, 206–227 (1985)

75. Portnoy, L., Eskin, E., Stolfo, S.J.: Intrusion detection with unlabeled data using clustering. In: Proceedings of ACM CSS Workshop on Data Mining Applied to Security (November 2001)
76. Pukkawanna, S., Fukuda, K.: Combining sketch and wavelet models for anomaly detection. In: 2010 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 313–319 (August 2010)
77. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.* 29(2), 427–438 (2000)
78. Resnick, S.: *Adventures in Stochastic Processes*. Birkhäuser (2002)
79. Ricciato, F., Coluccia, A., D’Alconzo, A., Veitch, D., Borgnat, P., Abry, P.: On the role of flows and sessions in internet traffic modeling: an explorative toy-model. In: *IEEE Globecom* (2009)
80. Ricciato, F., Coluccia, A., D’Alconzo, A.: A review of dos attack models for 3g cellular networks from a system-design perspective. *Computer Communications* 33(5), 551–558 (2010)
81. Ringberg, H., Soule, A., Rexford, J., Diot, C.: Sensitivity of pca for traffic anomaly detection. *SIGMETRICS Perform. Eval. Rev.* 35(1), 109–120 (2007)
82. Roughan, M., Veitch, D., Abry, P.: Real-time estimation of the parameters of long-range dependence. *IEEE/ACM Trans. Netw.* 8(4), 467–478 (2000)
83. Schweller, R., Gupta, A., Parsons, E., Chen, Y.: Reversible sketches for efficient and accurate change detection over network data streams. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement, IMC 2004, pp. 207–212. ACM, New York (2004)
84. Shiryaev, A.: On optimum methods in quickest detection problems. *Theory Probab. Appl.* 8, 22–46 (1963)
85. Shiryaev, A.: On Markov sufficient statistics in non-additive Bayes problems of sequential analysis. *Theory Probab. Appl.* 9, 604–618 (1964)
86. Shlens, J.: A tutorial on principal component analysis (December 2005), <http://www.sn1.salk.edu/~shlens/pub/notes/pca.pdf>
87. Shyu, M., Chen, S., Sarinapakorn, K., Chang, L.: A novel anomaly detection scheme based on principal component classifier. In: *IEEE Foundations and New Directions of Data Mining Workshop, in Conjunction with ICDM 2003*, pp. 172–179 (2003)
88. Siegmund, D.: *Sequential Analysis*. Springer (1985)
89. Sperotto, A., Mandjes, M., Sadre, R., de Boer, P.T., Pras, A.: Autonomic parameter tuning of anomaly-based IDSs: an SSH case study. *IEEE Transactions on Network and Service Management* 9, 128–141 (2012)
90. Subhabrata, B.K., Krishnamurthy, E., Sen, S., Zhang, Y., Chen, Y.: Sketch-based change detection: Methods, evaluation, and applications. In: *Internet Measurement Conference*, pp. 234–247 (2003)
91. Svoboda, P., Ricciato, F., Hasenleithner, E., Pilz, R.: Composition of gprs/umts traffic: snapshots from a live network. In: *4th Intl Workshop on Internet Performance, Simulation, Monitoring and Measurement, IPS-MOME 2006*, Salzburg (2006)
92. Tartakovsky, A., Veeravalli, V.: Change-point detection in multi-channel and distributed systems with applications. In: *Applications of Sequential Methodologies*, pp. 331–363 (2004)
93. Thorup, M., Zhang, Y.: Tabulation based 4-universal hashing with applications to second moment estimation. In: *SODA 2004: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 615–624. Society for Industrial and Applied Mathematics, Philadelphia (2004)
94. Thottan, M., Ji, C.: Anomaly detection in IP networks. *IEEE Trans. on Signal Processing* 51(8) (August 2003)

95. Thottan, M., Liu, G., Ji, C.: Anomaly detection approaches for communication networks. In: Cormode, G., Thottan, M., Sammes, A.J. (eds.) *Algorithms for Next Generation Networks. Computer Communications and Networks*, pp. 239–261. Springer, London (2010)
96. Tolle, J., Niggemann, O.: *Supporting intrusion detection by graph clustering and graph drawing*. Springer (2000)
97. Traynor, P., McDaniel, P., La Porta, T.: On attack causality in internet-connected cellular networks. In: *USENIX Security (August 2007)*
98. Traynor, P., McDaniel, P., La Porta, T.: *Security for Telecommunications Networks*. Springer (2008)
99. Vetterli, M., Kovačević, J.: *Wavelets and subband coding*. Prentice-Hall, Inc., Upper Saddle River (1995)
100. Wang, L., Potzelberger, K.: Boundary crossing probability for Brownian motion and general boundaries. *J. Appl. Probab.* 34, 54–65 (1997)
101. Wang, W., Battiti, R.: Identifying intrusions in computer networks with principal component analysis. In: *ARES 2006: Proceedings of the First International Conference on Availability, Reliability and Security*, pp. 270–279. IEEE Computer Society, Washington, DC (2006)
102. Wang, W., Guan, X., Zhang, X.: A Novel Intrusion Detection Method Based on Principle Component Analysis in Computer Security. In: Yin, F.-L., Wang, J., Guo, C. (eds.) *ISNN 2004, Part II. LNCS*, vol. 3174, pp. 657–662. Springer, Heidelberg (2004)
103. Yang, H., Ricciato, F., Lu, S., Zhang, L.: Securing a wireless world. *Proceedings of the IEEE* 94(2), 442–454 (2006)
104. Ye, N.: A markov chain model of temporal behavior for anomaly detection. In: *Proceedings of the Workshop on Information Assurance and Security (2000)*

Changepoint Detection Techniques for VoIP Traffic

Michel Mandjes¹ and Piotr Żuraniewski^{1,2,3}

¹ University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands

² AGH University of Science and Technology, Kraków, Poland

³ TNO, Delft, The Netherlands

Abstract. The control of communication networks critically relies on procedures capable of detecting unanticipated load changes. In this chapter we present an overview of such techniques, in a setting in which each connection consumes roughly the same amount of bandwidth (with VoIP as a leading example). For the situation of exponential holding times an explicit analysis can be performed in a large-deviations regime, leading to approximations of the test statistic of interest (and, in addition, to results for the transient of the M/M/∞ queue, which are of independent interest). This procedure being applicable to exponential holding times only, and also being numerically rather involved, we then develop an approximate procedure for general holding times. In this procedure we record the number of trunks occupied at equidistant points in time $\Delta, 2\Delta, \dots$, where Δ is chosen sufficiently large to safely assume that the samples are independent; this procedure is backed by results on the transient of the M/G/∞ queue. The validity of the testing procedures is demonstrated through set of numerical experiments; it is also pointed out how diurnal patterns can be dealt with. An experiment with real data illustrates the proposed techniques.

Keywords: Overload detection, infinite-server queues, transient, VoIP.

1 Introduction

When sizing communication networks, probably still the most frequently used tool is the celebrated *Erlang loss formula*, dating back to the early 1900s. This formula was originally developed for computing the blocking probability for circuit-switched (for instance voice) traffic sharing a trunk group of size, say $C \in \mathbb{N}$, and is often used for *dimensioning* purposes: it enables the selection of a value of C such that the blocking probability is below some tolerable level ε . Despite the fact that the formula has been around for a rather long time, it is still a cornerstone when resolving dimensioning issues, owing to its general applicability and its explicit, manageable form. Also, notice that it can in principle be used in any setting in which each connection requires (roughly) the same amount of bandwidth. As a consequence, it is also applicable in non-circuit-switched technologies, e.g. when considering voice-over-IP (VoIP).

In more detail, the Erlang loss formula is based on the (realistic) assumption of a Poisson arrival stream of flows (say, with intensity λ , expressed in Hertz or

s^{-1}). The call durations are independent and identically distributed, with mean $1/\mu$ (in s), and the load ρ is defined as the unit-less number $\rho := \lambda/\mu$. If there are C lines available, the probability of blocking in this model is

$$p(C | \rho) := \left(\frac{\rho^C}{C!} \right) / \left(\sum_{c=0}^C \frac{\rho^c}{c!} \right).$$

Importantly, this formula shows that for dimensioning the trunk group, no information on λ and μ is needed apart from their ratio $\rho = \lambda/\mu$. Observe that no assumption on the distribution of the call holding times was imposed; the above formula applies for all holding-time distributions with mean $1/\mu$. We denote by $\bar{\rho}$ the maximum load ρ such that $p(C | \rho)$ is below some predefined tolerance ε . The underlying queueing model is often referred to as the M/G/C/C queue; a useful approximation of $p(C | \rho)$ is the probability that the number of busy servers in the corresponding infinite-server queue (i.e., M/G/ ∞) exceeds C .

When operating a network one has to constantly check the validity of the input assumptions the dimensioning decision was based upon. More concretely, one has to check whether the load ρ has not reached the maximum allowable load $\bar{\rho}$. Clearly, if the load has increased beyond $\bar{\rho}$, measures have to be taken to deal with the overload, perhaps by rerouting the excess calls, or, on a longer timescale, by increasing the available capacity.

This motivates why it is of crucial importance to design procedures to (statistically) assess whether the load has changed. In statistical terms we would call this a ‘change-point detection problem’ [11]: from observations of the number of lines used, we wish to infer whether a load change has taken place. Also, one would like to know *when* the change has occurred; then an alarm can be issued that triggers traffic management measures (overload control, such as rerouting, or temporary adaptations of the amount of bandwidth available to the calls).

Empirical guidelines for the problem described above have been developed in e.g., [5], but there is a clear need for more rigorously supported procedures. Without aiming to give an exhaustive overview, we mention here related work on a fractal model [14], and also [3, 13]. An application of the celebrated CUSUM technique [11] in the networking domain can be found in [7]. Several valuable contributions to the change-point detection problem are due to Tartakovsky and co-authors, cf. [12].

The high-level goal of this chapter is to sketch the state-of-the-art in this field, summarizing the material developed in [8, 9, 15]. The main messages of the present chapter are the following. (i) We first consider the case in which the call durations have an exponential distribution. We show how a likelihood-based CUSUM-type of test can be set up. The crucial complication is that the number of trunks occupied does *not* constitute a sequence of i.i.d. random variables (as there will be dependence between subsequent observations). Therefore the ‘traditional’ CUSUM result does not apply here, and a new approach had to be developed. Setting up our test requires knowledge of the transient probabilities in the corresponding M/M/ ∞ system. We first show how, in a large-deviations setting, these transient probabilities can be determined. These have interesting features, such as a so-called bifurcation, as in [10]. The test also requires the

computation of the probability that a sum of likelihoods exceeds some threshold. We show how this can be done, relying on calculus-of-variations techniques.

(ii) The findings above being only applicable to the case of exponentially distributed call durations, and given the high numerical complexity of the resulting procedure, we then look for an approach that works for the $M/G/\infty$ in general, and that requires substantially less computational effort. We explain how we can use classical changepoint-detection, which rely on the assumption of independent observations (where the observations correspond to samples of the number of calls in progress, at equidistant points in time, say $\Delta, 2\Delta, \dots$). This independence assumption is clearly not fulfilled in our model, at least not formally, but evidently for Δ sufficiently large the dependence will have a minor impact. We develop new estimates on the relaxation time of the $M/G/\infty$ queue, which tell us how large Δ should be in order to be able to safely assume independence.

(iii) The third contribution is that we show how accurately the proposed procedures can detect overload. This we do through a series of simulation experiments and experiments with real traces. Special attention is paid to the trade-off between the detection ratio and the false alarm rate. The experiments indicate that our procedure, after some tuning, provides a powerful technique for changepoint detection.

A remark is in place. A substantial part of the material presented in this chapter relies on the assumption that the load value in the ‘baseline model’ is (roughly) constant in time; the idea is that we would like to detect deviations from this stationary pattern. It is evident that in practice such a stationarity assumption does not apply. Essentially two remedies can be thought of. The first is that the day pattern is split into intervals in which the load is more or less constant (but the price to be paid is that these periods are relatively short, viz. up to a few hours at most); this approach was followed in [8]. The second approach is to apply a trend removal procedure, as in [9]; basically, the approach presented filters out the diurnal pattern, so that we obtain, after a normalization, (approximately) standard Normal residuals.

This chapter is organized as follows. In Section 2 we present our model and some preliminaries, and define our goal in terms of a changepoint detection problem. Section 3 presents a framework for changepoint detection for the $M/M/\infty$ model, whereas Section 4 presents the approximate analysis for the $M/G/\infty$ model. The last sections are devoted to numerical experimentation, both in a simulation-based setting and by using real traffic traces; a short description of a trend removal procedure is presented as well.

2 Model, Preliminaries, and Goals

In this section we describe the goals of the chapter, and the underlying mathematical model. Our analysis will be based on the $M/G/\infty$ queue, that is, a service system in which calls arrive according to a Poisson process (with rate, say, λ), where it is assumed that the call durations form an i.i.d. sequence B_1, B_2, \dots , and infinitely many servers. With $1/\mu$ denoting the mean value of a generic call duration B , the load of the system is defined as $\rho := \lambda/\mu$. It is well-known that

the stationary distribution of the number of calls simultaneously present, say Y , is Poisson with mean ρ . Also the transient distribution of this system can be dealt with fairly explicitly. Suppose that $Y(t)$ denotes the number of trunks occupied at time t , and assuming that the queue is in stationarity at time 0, the following decomposition applies. Conditioning on $Y(0) = k$, with ‘=d’ denoting equality in distribution, we have that

$$Y(t) =_d \text{Bin}(k, p_t) + \text{Pois}(\lambda t q_t), \tag{1}$$

where $\text{Bin}(k, p)$ denotes a binomial random variable with parameters k and p , and $\text{Pois}(\lambda)$ as Poisson random variable with mean λ ; in addition, the binomial and Poisson random variables in the right-hand side of (1) are independent. Here, p_t is the probability that an arbitrary call that is present at time 0 is still present at time t , which equals $\mathbb{P}(B^* > t) = \mathbb{E}B^{-1} \int_t^\infty \mathbb{P}(B > s) ds$, where B^* denotes the excess life-time distribution of B . Likewise, q_t is the probability that an arbitrary call that arrives in $(0, t]$ is still present at time t ; using the fact that the arrival epoch of such an arbitrary call is uniformly distributed on $(0, t]$, conditioning on the arrival epoch $s \in (0, t]$ yields that

$$q_t = t^{-1} \int_0^t \mathbb{P}(B > t - s) ds = t^{-1} \int_0^t \mathbb{P}(B > s) ds = \mathbb{E}B \cdot t^{-1} \cdot \mathbb{P}(B^* \leq t).$$

Observe that the mean of the Poissonian term in the right-hand side of (1), $\lambda t q_t$, equals $\rho \mathbb{P}(B^* < t)$. It is readily verified that the correlation coefficient of $Y(0)$ and $Y(t)$ equals $\text{Corr}(Y(0), Y(t)) = \mathbb{P}(B^* > t)$; here it is used that $Y(0)$ has a Poisson distribution with mean ρ .

As mentioned in the introduction, the goal of the chapter is to detect changes in the load imposed on a M/G/ ∞ queue. More specifically, with ρ the load imposed on the queueing resource, and $\bar{\rho}$ the maximum allowable load (in order to meet a given performance criterion, for instance in terms of a blocking probability), we want to test whether all samples correspond to load ρ (which we associate with hypothesis H_0), or whether there has been a changepoint within the data set, such that before the changepoint the data were in line with load ρ , and after the changepoint with $\bar{\rho}$ (which is hypothesis H_1).

3 Analysis for M/M/ ∞

In this section we consider the case that the calls are i.i.d. samples from an exponential distribution with mean $1/\mu$; the model is then known as M/M/ ∞ . We consider the discrete-time Markovian model describing the dynamics of the number of trunks occupied, by recording the continuous-time process at the embedded epochs at which this number changes.

Let, for $i = 1, 2, \dots$, $Y_i := \sum_{j=1}^i X_j$, where the probabilities $\mathbb{P}(X_i = \pm 1 \mid Y_{i-1})$ are defined through, for given numbers λ_m and μ_m ,

$$(X_i \mid Y_{i-1} = m) = \begin{cases} 1 & \text{with probability } \lambda_m \\ -1 & \text{with probability } \mu_m = 1 - \lambda_m. \end{cases}$$

As mentioned above, in this section we assume that the dynamics of the number of trunks occupied are described by the M/M/∞ model, i.e., $\lambda_m \equiv \lambda_m(\varrho) = \lambda/(\lambda + m\mu) = \varrho/(\varrho + m)$, with $\varrho := \lambda/\mu$. We consider the model with an infinite number of trunks available; then the (steady-state) probability of C calls present can be used as an approximation of the blocking probability in the model with C lines.

In this section, our analysis relies on applying the so-called *many-flows scaling*. Under this scaling the load is renormalized by n (that is, we replace $\varrho \mapsto n\varrho$), and at the same time the number of trunks is inflated by a factor n , as motivated in [10, Ch. 12]. It effectively means that we can use *large-deviations theory* to asymptotically (large n) determine the distribution of the number of calls simultaneously present. Under this scaling the steady-state number of calls present has a Poisson distribution with mean $n\varrho$, i.e., $\mathbb{P}(Y = k) = (n\varrho)^k e^{-n\varrho}/k!$. A straightforward application of Stirling’s formula yields the following expression for the exponential decay rate of $\mathbb{P}(Y = \lfloor n\beta \rfloor)$:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}(Y = \lfloor n\beta \rfloor) = -\varrho + \beta + \beta \log \left(\frac{\varrho}{\beta} \right) =: \xi(\beta);$$

here we recognize the large-deviations rate function of the Poisson distribution [10, Example 1.13]. Using Cramér’s theorem, we also have that the probability $\mathbb{P}(Y \geq n\beta)$ has the same exponential decay rate.

Goal: changepoint. We want to test whether there is a ‘changepoint’, that is, during our observation period the load parameter ϱ (which we let correspond to the probability model \mathbb{P}) changes into $\bar{\varrho} \neq \varrho$ (the model \mathbb{Q}). More formally, we consider the following (multiple) hypotheses.

- H_0 : $(X_i)_{i=1}^n$ is distributed according to the above described birth-death chain with parameter ϱ .
- H_1 : For some $\delta \in \{1/n, 2/n, \dots, (n-1)/n\}$, it holds that $(X_i)_{i=1}^{\lfloor n\delta \rfloor}$ is distributed according to the birth-death chain with parameter ϱ , whereas $(X_i)_{i=\lfloor n\delta \rfloor+1}^n$ is distributed according to the birth-death chain with parameter $\bar{\varrho} \neq \varrho$.

Inspired by the Neyman-Pearson lemma, see e.g. [2, Ch. V.E and Appendix E], we consider the following likelihood-ratio test statistic:

$$T_n := \max_{\delta \in [0,1)} T_n(\delta), \text{ with } T_n(\delta) := \frac{1}{n} \sum_{i=\lfloor n\delta \rfloor+1}^n L_i - \varphi(\delta), \quad L_i := \log \frac{\mathbb{Q}(X_i | Y_{i-1})}{\mathbb{P}(X_i | Y_{i-1})},$$

for some function $\varphi(\cdot)$ we will specify later. If $T_n > 0$, then H_0 is rejected.

To enable statistical tests, we wonder what the probability is, under H_0 , that the above test statistic is larger than 0. For reasons of tractability, we consider its exponential decay rate (asymptotic in the scaling parameter n):

$$\eta(\varphi) := \lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}(T_n > 0), \quad \eta(\varphi | \beta_0) := \lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}(T_n > 0 | Y_0 = n\beta_0)$$

These decay rates can be evaluated as follows. We define $\eta(\varphi, \delta \mid \beta_0)$, $\bar{\eta}(\varphi, \delta \mid \beta_\delta)$, and $\xi(\beta_\delta \mid \beta_0)$, as the exponential decay rates of, respectively,

$$\mathbb{P}(T_n > 0 \mid Y_0 = n\beta_0), \quad \mathbb{P}(T_n > 0 \mid Y_{\lfloor n\delta \rfloor} = n\beta_\delta), \quad \mathbb{P}(Y_{\lfloor n\delta \rfloor} = n\beta_\delta \mid Y_0 = n\beta_0).$$

Standard large-deviations argumentation yields that

$$\eta(\varphi \mid \beta_0) = \sup_{\delta \in [0,1]} \eta(\varphi, \delta \mid \beta_0) = \sup_{\delta \in [0,1]} \sup_{\beta_\delta > 0} (\xi(\beta_\delta \mid \beta_0) + \bar{\eta}(\varphi, \delta \mid \beta_\delta))$$

(‘principle of the largest term’), and also

$$\eta(\varphi) = \sup_{\delta \in [0,1]} \sup_{\beta_\delta > 0} (\xi(\beta_\delta) + \bar{\eta}(\varphi, \delta \mid \beta_\delta)).$$

The decay rate of the transient probabilities, that is, $\xi(\beta_\delta \mid \beta_0)$, will be analyzed in Section 3.1, and the decay rate of the exceedance probabilities $\bar{\eta}(\varphi, \delta \mid \beta_\delta)$ (which we will sometimes refer to as ‘likelihood probabilities’) in Section 3.2.

3.1 Transient Probabilities

To analyze the decay rate of $\mathbb{P}(Y_{\lfloor n\delta \rfloor} = n\beta_\delta \mid Y_0 = n\beta_0)$, we rely on *Slow Markov Walk* theory [2, Ch. IV.C]. As this technique has been described in detail in [2] we restrict ourselves to sketching the main steps. Then we show how to apply this theory to determine the transient probabilities $\xi(\beta_\delta \mid \beta_0)$.

Slow Markov Walk. A prominent role in Slow Markov Walk theory is played by the so-called ‘local large deviations rate function’ $I_x(u)$, defined as

$$\sup_{\theta} \left(\theta u - \log \left(e^\theta \lambda_{nx}(n\varrho) + e^{-\theta} (1 - \lambda_{nx}(n\varrho)) \right) \right) = \sup_{\theta} \left(\theta u - \log \left(\frac{\varrho e^\theta + x e^{-\theta}}{\varrho + x} \right) \right).$$

Intuitively reasoning, $I_x(u)$ measures the ‘effort the process has to make’ (per time unit), starting in state x , to move into direction u . It follows that

$$\theta^* \equiv \theta_x^*(u) = \frac{1}{2} \log \left(\frac{x}{\varrho} \cdot \frac{1+u}{1-u} \right), \tag{2}$$

θ^* denoting the optimizing θ ; if θ^* is positive (negative) the process has to ‘speed up’ (‘slow down’) to be moving into direction u . Slow Markov Walk theory yields the exponential decay rates of the empirical mean process $n^{-1} \cdot Y_{\lfloor nt \rfloor}$ to be in a certain set, or close to a given function f . Loosely speaking, it says that

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P} \left(\frac{1}{n} \cdot Y_{\lfloor nt \rfloor} \approx f(t), t \in [0, \delta] \right) = - \int_0^\delta I_{f(t)}(f'(t)) dt;$$

sometimes the right-hand side of the previous display is referred to as the ‘cost’ of the path f in the interval $[0, \delta]$. In this sense, we can determine also the ‘average path’ of $Y_{\lfloor nt \rfloor}/n$, which is the path with *zero* cost: it consists of pairs

$(f(t), f'(t))$ for which $I_{f(t)}(f'(t)) = 0$, or, put differently, $\theta_{f(t)}^*(f'(t)) = 0$. It can be calculated that this average path is given through the differential equation

$$\frac{f(t)}{\varrho} \cdot \frac{1 + f'(t)}{1 - f'(t)} = 1, \quad \text{or} \quad f'(t) = \frac{\varrho - f(t)}{\varrho + f(t)}; \quad (3)$$

This path converges to the ‘mean’ $f(\infty) = \varrho$ as $t \rightarrow \infty$, as was expected. Inserting θ^* , as given in (2), we find after tedious computations that $I_x(u)$ equals

$$\frac{1}{2}u \log \left(\frac{1+u}{1-u} \right) + \frac{1}{2}u \log \left(\frac{x}{\varrho} \right) - \frac{1}{2} \log \left(\frac{x\varrho}{(\varrho+x)^2} \right) - \log 2 + \frac{1}{2} \log(1-u^2).$$

Determining the decay rate of $\xi(\beta_\delta | \beta_0)$. We now reduce the search for the decay rate $\xi(\beta_\delta | \beta_0)$ to a variational problem. Slow Markov Walk theory says that

$$\xi(\beta_\delta | \beta_0) = - \inf_{f \in \mathcal{A}} \int_0^\delta I_{f(t)}(f'(t)) dt,$$

where the set \mathcal{A} consists of all paths f such that $f(0) = \beta_0$ and $f(\delta) = \beta_\delta$. This variational problem can be solved by applying elementary results from calculus of variations; see for instance [10, Appendix C]. The optimizing path is characterized by the so-called DuBois-Reymond equation [10, Eq. (C.3)]:

$$I_{f(t)}(f'(t)) - f'(t) \cdot \frac{\partial}{\partial u} I_x(u) \Big|_{x=f(t), u=f'(t)} = K,$$

or, equivalently,

$$\log \left((1 - (f'(t))^2) \cdot \frac{(\varrho + f(t))^2}{4f(t)\varrho} \right) = K;$$

the K is to be determined later on (and essentially serves as a ‘degree-of-freedom’, to be chosen such that $f(\delta) = \beta_\delta$). After some elementary algebraic manipulations we find the ordinary differential equation

$$f'(t) = \pm \sqrt{1 - e^K \cdot \frac{4f(t)\varrho}{(\varrho + f(t))^2}}; \quad (4)$$

notice that for $K = 0$ we retrieve the ‘average path’ (3), as expected.

Interestingly, we can explicitly find the inverse of the solution of (4) (that is, t in terms of f , rather than f in terms of t). Recalling that $\varrho + f(t) > 0$, by separating variables we obtain

$$t = \pm \int \frac{(\varrho + f)}{\sqrt{f^2 + (2\varrho - 4\varrho e^K)f + \varrho^2}} df.$$

With $b_\varrho := 2\varrho - 4\varrho e^K$, standard calculus eventually gives that $t \equiv t(f)$ equals

$$\pm \left(\sqrt{f^2 + b_\varrho f + \varrho^2} + 2\varrho e^K \log \left(f + \varrho - 2\varrho e^K + \sqrt{f^2 + b_\varrho f + \varrho^2} \right) + \gamma \right), \quad (5)$$

where γ is chosen such that the boundary condition, i.e., $f(0) = \beta_0$, is met.

Numerical Evaluation. To obtain path the $f(t)$, for a given value of K , (5) needs to be solved, but obviously there are alternatives. One could for instance solve the differential equation (4) iteratively starting in $f(0) = \beta_0$, by applying techniques of the Runge-Kutta type. This is not entirely standard, though, as the path may be horizontal at some point between 0 and δ (so that the most straightforward numerical procedures do not work).

A second step the is then to find a value of K such that indeed $f_K^*(\delta) = \beta_\delta$. Notice that, because we can move up or down by just 1, we have to require that $\beta_\delta \in [\max\{0, \beta_0 - \delta\}, \beta_0 + \delta]$. Once we have found the optimal path (having taken into account the condition $f_K^*(\delta) = \beta_\delta$), say the path $f^*(\cdot)$, we can (numerically) evaluate

$$\int_0^\delta I_{f^*(t)}((f^*)'(t))dt,$$

thus finding the decay rate $\xi(\beta_\delta | \beta_0)$.

As indicated, we proceed by making a few observations that enable the numerical evaluation of the decay rate $\xi(\beta_\delta | \beta_0)$.

- If $\beta_0 < \varrho$ and $\beta_\delta > \varrho$ or vice versa the above differential equation can, for any given K , be numerically solved in a straightforward fashion, because the path will be *monotone*. More precisely, one can rely on well-known Runge-Kutta techniques, starting in $f(0) = \beta_0$. By varying the value of K , we can then find the path that is at β_δ at time δ .

Analysis analogous to [10, Ch. 12] reveals the following properties. (i) Suppose $\beta_0 < \varrho < \beta_\delta$. Then the K that is such that $f_K(\delta) = \beta_\delta$ is *negative*. The above iterative Runge-Kutta scheme can be used, with for instance a bisection loop that selects the right $K < 0$. (ii) If $\beta_\delta < \varrho < \beta_0$, the optimal path is the time-reversed of the path that starts in β_δ and ends in β_0 . This means that the optimal path can be identified as under (i), i.e., starting in $\beta_\delta < \varrho$, and ending in $\beta_0 > \varrho$; note that the decay rate differs, though (but can be determined by numerically evaluating the integral over the local rate function along the resulting path).

- Problems may arise, however, when the optimal path may have derivative 0 at some point in $(0, \delta)$. This is typically the case when β_0 and β_δ or both smaller or larger than ϱ , and δ is at the same time relatively large (as then the optimal path is such that the number of trunks occupied, starting from β_0 , is first ‘pulled’ towards ϱ , and then ‘pushed back’ into the direction of β_δ). Interestingly, for given $K > 0$, one can compute the value f_K of $f(s)$ at the point s for which $f'(s) = 0$. It turns out that

$$f_K = \varrho \left(2e^K - 1 \pm 2\sqrt{e^{2K} - e^K} \right);$$

elementary arguments show that we have to take the --sign (+-sign) when β_0 and β_δ are both smaller (larger) than ϱ . Also, it is readily verified that for $K = 0$ one obtains $f_K = \varrho$, and for $K \rightarrow \infty$ in the --branch $f_K \rightarrow 0$

and in the +-branch $f_K \rightarrow \infty$. The solution has, as in [10, Section 12.5], a *bifurcation point*: for small δ (say, δ smaller than some critical timescale T) the path will typically be monotone ($K < 0$), whereas for larger δ (i.e., $\delta > T$) the path will have slope 0 for some point between 0 and δ ($K > 0$). There is no explicit expression for the timescale T available, but we can identify a timescale $T^- < T$ such that for any smaller δ the path will be monotone, as follows.

First observe that we can explicitly solve (3) to obtain, for a constant γ :

$$t = \pm(-2\varrho \log(\varrho + f(t)) - f(t) + \gamma);$$

unfortunately we cannot invert this relation (thus obtaining $f(t)$ as function of t explicitly). Mimicking the argumentation in [10, Section 12.5], we find T^- by imposing $f(T^-) = \beta_0$, while γ is determined through $f(0) = \beta_\delta$ (here, again, time-reversibility properties are applied). We thus arrive at, for obvious reasons using the absolute value,

$$T^- = \left| 2\varrho \log\left(\frac{\varrho - \beta_0}{\varrho - \beta_\delta}\right) + \beta_0 - \beta_\delta \right|.$$

We arrive at the following conclusion:

- For $\delta < T^-$ the path is monotone, we have $K < 0$, and we can use the method described for the case $\beta_0 < \varrho < \beta_\delta$. Use the +-branch of the differential equation.
- For $\delta > T^-$ one should realize that $\delta > T^-$ is just a necessary but not sufficient condition for a non-monotone optimal path to occur, as argued in [10, Section 12.5]; for $\delta \in [T^-, \infty)$ close to T^- still monotone paths come out. The bifurcation point T can be determined empirically.

3.2 Likelihood Probabilities

In this section we analyze the decay rate $\bar{\eta}(\varphi, \delta \mid \beta_\delta)$, using the same methodology as in Section 3.1. As the line of reasoning is very similar to the one followed in Section 3.1, we just sketch the basic steps.

First observe that we can shift time so that we obtain

$$\bar{\eta}(\varphi, \delta \mid \beta_\delta) = \lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}(\bar{T}_n(\delta) > 0 \mid Y_0 = n\beta_\delta), \bar{T}_n(\delta) := \frac{1}{n} \sum_{i=1}^{\lfloor n(1-\delta) \rfloor} L_i - \varphi(1-\delta);$$

if this is indeed a large deviation probability, then we can replace the inequality ' $> \varphi(1 - \delta)$ ' by an equality ' $= \varphi(1 - \delta)$ '. We again want to use Slow Markov Walk theory, in that we wish to evaluate

$$\bar{\eta}(\varphi, \delta \mid \beta_\delta) = - \inf_{f \in \mathcal{B}} \int_0^{1-\delta} I_{f(t)}(f'(t)) dt,$$

where \mathcal{B} are the paths (with $f(0) = \beta_\delta$) such that $\lim_{n \rightarrow \infty} n^{-1} \cdot Y_{[nt]} = f(t)$, for $t \in [0, \delta)$, implies that $\lim_{n \rightarrow \infty} \bar{T}_n(\delta) = \varphi(1 - \delta)$. Let us characterize the paths with this property. To this end, first rewrite

$$g_f(\delta) := \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^{\lfloor n(1-\delta) \rfloor} L_i = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^{(1-\delta)/\varepsilon} \sum_{i=n(k-1)\varepsilon+1}^{nk\varepsilon} L_i.$$

For i in $\{n(k-1)\varepsilon+1, \dots, nk\varepsilon\}$ we have that

$$\frac{\mathbb{Q}(X_i | Y_{i-1})}{\mathbb{P}(X_i | Y_{i-1})} = \frac{\bar{\varrho}}{\varrho} \cdot \frac{\varrho + f(k\varepsilon)}{\bar{\varrho} + f(k\varepsilon)} + O(\varepsilon) \quad \text{and} \quad \frac{\mathbb{Q}(X_i | Y_{i-1})}{\mathbb{P}(X_i | Y_{i-1})} = \frac{\varrho + f(k\varepsilon)}{\bar{\varrho} + f(k\varepsilon)} + O(\varepsilon)$$

if $X_i = 1$ and $X_i = -1$, respectively. Let $U_{k,n}$ be the number of steps upwards in $\{n(k-1)\varepsilon+1, \dots, nk\varepsilon\}$, and $D_{k,n}$ the number of steps downwards. Then trivially $U_{k,n} + D_{k,n} = n\varepsilon$, but on the other hand $U_{k,n} - D_{k,n} = n\varepsilon f'(k\varepsilon) + O(\varepsilon^2)$. From these relations we can solve $U_{k,n}$ and $D_{k,n}$. We end up with

$$\sum_{k=1}^{(1-\delta)/\varepsilon} \left(\frac{\varepsilon}{2} f'(k\varepsilon) \log \left(\frac{\bar{\varrho}}{\varrho} \right) - \frac{\varepsilon}{2} \log \left(\frac{\bar{\varrho}}{\varrho} \right) + \varepsilon \log \left(\frac{\varrho + f(k\varepsilon)}{\bar{\varrho} + f(k\varepsilon)} \right) + O(\varepsilon^2) \right).$$

Letting $\varepsilon \downarrow 0$, we obtain

$$\begin{aligned} g_f(\delta) &= \int_0^{1-\delta} \frac{1}{2} \log \left(\frac{\bar{\varrho}}{\varrho} \right) \cdot f'(t) dt - \frac{1-\delta}{2} \log \left(\frac{\bar{\varrho}}{\varrho} \right) + \int_0^{1-\delta} \log \left(\frac{\varrho + f(t)}{\bar{\varrho} + f(t)} \right) dt \\ &= h_f(\delta) + \int_0^{1-\delta} \log \left(\frac{\varrho + f(t)}{\bar{\varrho} + f(t)} \right) dt, \end{aligned}$$

with $h_f(\delta) := \frac{1}{2} \log(\bar{\varrho}/\varrho) \cdot (f(1-\delta) - f(0)) - \frac{1}{2}(1-\delta) \log(\bar{\varrho}/\varrho)$. Hence we are left with a variational problem, with Lagrange multiplier L :

$$\inf_{f \in \mathcal{B}} \left(\int_0^{1-\delta} \left(I_{f(t)}(f'(t)) - L \log \left(\frac{\varrho + f(t)}{\bar{\varrho} + f(t)} \right) \right) dt - L h_f(\delta) \right). \tag{6}$$

The DuBois-Reymond equation reads

$$I_{f(t)}(f'(t)) - f'(t) \cdot \frac{\partial}{\partial u} I_x(u) \Big|_{x=f(t), u=f'(t)} - L \log \left(\frac{\varrho + f(t)}{\bar{\varrho} + f(t)} \right) = K,$$

which reduces to

$$f'(t) = \pm \sqrt{1 - e^K \left(\frac{\varrho + f(t)}{\bar{\varrho} + f(t)} \right)^L \frac{4f(t)\varrho}{(\varrho + f(t))^2}}.$$

We again need to numerically solve this, under $f(0) = \beta_\delta$. K and L should be chosen such that $g_f(\delta) = \varphi(1 - \delta)$ and (6) is minimal. In more detail, a procedure could be the following. For given K, L , solve the differential equation,

to obtain the path $f_{K,L}^*(\cdot)$. For given L , determine the $K \equiv K(L)$ such that $gf_{K,L}^*(\delta) = \varphi(\delta)$. Then minimize, over L ,

$$\int_0^{1-\delta} I_{f_{K(L),L}^*(t)}((f_{K(L),L}^*)'(t))dt.$$

It is clear, however, that such procedures are, from a numerical standpoint, in general quite involved. A substantial simplification can be achieved by approximating the functions involved by polynomial functions (cf. Ritz method).

3.3 Discussion

Now that we have derived in Sections 3.1 and 3.2 expressions for the decay rate of interest, it remains to select an appropriate function $\varphi(\cdot)$. We can choose, for a given value of β_0 , $\varphi(\cdot)$ such that $\eta(\varphi, \delta | \beta_0) \equiv \alpha$ for all $\delta \in [0, 1)$. As argued in [2, Ch. V.E], this choice gives the best type-II error rate performance.

The procedure described above is a natural counterpart for the ‘usual’ changepoint detection procedures that were designed for i.i.d. increments; importantly, we recall the fact that in our model the increments are dependent made it necessary to develop a new method. The most significant drawbacks of the above procedure are: (i) it only applies to the case of exponentially distributed call durations; (ii) its computational complexity is high. In the next section we present an approach with is somewhat more crude, but overcomes these two problems.

4 Analysis for M/G/ ∞

In this section we present an approach to do changepoint detection in an M/G/ ∞ queue. Clearly, the observations $Y(0), Y(\Delta), Y(2\Delta), \dots$ are *not* independent; remember from Section 2 that the correlation coefficient between $Y(0)$ and $Y(\Delta)$ is given by $\mathbb{P}(B^* > \Delta)$. It is evident, however, that this dependence is negligible for Δ sufficiently large. In Section 4.1 we analyze how large Δ should be to be able to safely assume independence – as a useful by-product, we derive insight into the so-called relaxation times in the M/G/ ∞ queue (which can be interpreted as a measure of the speed of convergence to the stationary distribution). Then Section 4.2 describes a changepoint detection procedure, which again relies on Slow Markov Walk theory [2, Ch. IV.C]; however, where we used this framework for *dependent* observations in Section 3, we now focus on the case in which the observations are i.i.d. (and sampled from a Poisson distribution).

4.1 Transient Probabilities

We first focus on the question: for a given number of calls present at time 0, how fast does the (transient) distribution of the number of calls present at time t , converge to the stationary distribution? This speed of convergence is often referred to as *relaxation time*, cf. [1]. We now identify $u_{k,\ell}(\cdot)$ such that

$$\lim_{t \rightarrow \infty} (u_{k,\ell}(t))^{-1} \cdot (\mathbb{P}(Y(t) = \ell | Y(0) = k) - \mathbb{P}(Y = \ell)) = 1. \quad (7)$$

We first observe that that, due to (1),

$$\mathbb{P}(Y(t) = \ell \mid Y(0) = k) = \sum_{m=0}^{\min\{k, \ell\}} \mathbb{P}(\text{Bin}(k, p_t) = m) \mathbb{P}(\text{Pois}(\lambda t q_t) = \ell - m).$$

Take the term corresponding to $m = 0$ in the summation in the right-hand side of the previous display, and subtract $\mathbb{P}(Y = \ell)$. We then obtain that $\varrho(r_{\ell-1} - r_\ell) \cdot \mathbb{P}(B^* > t) \cdot (1 + o(1))$ as $t \rightarrow \infty$, recalling that

$$\lambda t q_t = \varrho \mathbb{P}(B^* < t) = \varrho - \varrho \mathbb{P}(B^* > t)$$

and denoting $r_\ell := e^{-\varrho} \varrho^\ell / \ell!$; here we used that

$$\lim_{t \rightarrow \infty} \frac{f(\varrho) - f(\varrho(1 - \mathbb{P}(B^* > t)))}{\varrho \mathbb{P}(B^* > t)} = f'(\varrho).$$

The term corresponding to $m = 1$ obeys $kr_{\ell-1} \cdot \mathbb{P}(B^* > t) \cdot (1 + o(1))$. Finally observe that the terms corresponding to $m \geq 2$ are $o(\mathbb{P}(B^* > t))$. Combining the above findings, we conclude that (7) indeed applies, with

$$u_{k, \ell}(t) = U_{k, \ell} \cdot \mathbb{P}(B^* > t), \quad U_{k, \ell} := \varrho(r_{\ell-1} - r_\ell) + kr_{\ell-1}.$$

Suppose is our goal is to enforce ‘approximate independence’ between $Y(0)$ and $Y(t)$ by choosing t sufficiently large that for all $k, \ell \in \{0, \dots, C\}$ we have that $|U_{k, \ell}| \cdot \mathbb{P}(B^* > t) < \varepsilon_{\max}$. Observe that $U_{k, \ell} \leq (\varrho + k)r_{\ell-1}$. Now use that the mode of the Poisson distribution lies close to ϱ : for all $i = 0, 1, \dots$

$$r_i \leq g(\varrho) := r_{\varrho_m}, \quad \varrho_m := \lfloor \varrho \rfloor \text{ if } \varrho \notin \mathbb{N}, \text{ and } \varrho \text{ else.}$$

We conclude that $U_{k, \ell} \leq (\varrho + C)g(\varrho)$ for all $k, \ell \in \{0, \dots, C\}$. Likewise, $U_{k, \ell} \geq -\varrho r_\ell \geq -\varrho g(\varrho)$. It is now straightforward to choose t such that for all $k, \ell \in \{0, \dots, C\}$ it holds that $|U_{k, \ell}| \cdot \mathbb{P}(B^* > t) < \varepsilon_{\max}$.

4.2 Changepoint Detection Procedure

As described above, we can now choose Δ so large that $u_{k, \ell}(\Delta) < \varepsilon_{\max}$, for all $k, \ell \in \{1, \dots, C\}$ and ε_{\max} some given small positive number. In this way we enforced ‘approximate independence’, thus justifying the use of procedures for i.i.d. observations, as in [2, Section VI.E].

Goal: changepoint. Again, we wish to detect a changepoint, that is, during our observation period the load parameter ϱ (which we let again correspond to the probability model \mathbb{P}) changes into $\bar{\varrho} \neq \varrho$ (the model \mathbb{Q}). More formally, we consider the following (multiple) hypotheses. Let $Y_i := Y(i\Delta)$ be the sequence of observations of the number of calls present at time $i\Delta$.

H_0 : $(Y_i)_{i=1}^n$ are distributed $\mathbb{Pois}(\varrho)$.

H_1 : For some $\delta \in \{1/n, 2/n, \dots, (n-1)/n\}$, it holds that $(Y_i)_{i=1}^{\lfloor n\delta \rfloor}$ is distributed $\mathbb{Pois}(\varrho)$, whereas $(Y_i)_{i=\lfloor n\delta \rfloor+1}^n$ is distributed $\mathbb{Pois}(\bar{\varrho})$, with $\bar{\varrho} \neq \varrho$.

Again, in view of the Neyman-Pearson lemma, we consider the following likelihood-ratio test statistic: for some function $\varphi(\cdot)$ we will provide later,

$$T_n := \max_{\delta \in [0,1]} T_n(\delta), \text{ with } T_n(\delta) := \frac{1}{n} \sum_{i=\lfloor n\delta \rfloor + 1}^n L_i - \varphi(\delta),$$

$L_i := \log \mathbb{Q}(Y_i)/\mathbb{P}(Y_i) = (\varrho - \bar{\varrho}) + Y_i \log(\bar{\varrho}/\varrho)$. If T_n is larger than 0, we reject H_0 . We can now use the machinery of [2, Section VI.E] to further specify this test. We first introduce the moment generating function and its Legendre transform:

$$\begin{aligned} M(\vartheta) &= \sum_{k=0}^{\infty} \left(\frac{\bar{\varrho}^k}{k!} e^{-\bar{\varrho}} \right)^{\vartheta} \left(\frac{\varrho^k}{k!} e^{-\varrho} \right)^{1-\vartheta} = e^{-\varrho} e^{(\varrho - \bar{\varrho})\vartheta} \exp \left(\varrho \left(\frac{\bar{\varrho}}{\varrho} \right)^{\vartheta} \right); \\ I(u) &= \sup_{\vartheta} (\vartheta u - \log M(\vartheta)) = \vartheta^*(u) u - \log M(\vartheta^*(u)), \text{ where} \\ \vartheta^*(u) &:= \frac{\log(u + \bar{\varrho} - \varrho) - \log(\varrho \log(\bar{\varrho}/\varrho))}{\log(\bar{\varrho}/\varrho)}. \end{aligned}$$

From [2, Section VI.E, Eqn. (46)–(48)], we can compute the decay rate of issuing an alarm under H_0 , for a given threshold function $\varphi(\cdot)$:

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P} \left(\max_{\delta \in [0,1]} T_n(\delta) > 0 \right) &= \max_{\delta \in [0,1]} (1 - \delta) \cdot \lim_{n \rightarrow \infty} \frac{1}{n(1 - \delta)} \log \mathbb{P} \left(\frac{T_n(\delta)}{1 - \delta} > 0 \right) \\ &= \max_{\delta \in [0,1]} \psi(\delta) \text{ with } \psi(\delta) := (1 - \delta) \cdot I \left(\frac{\varphi(\delta)}{1 - \delta} \right). \end{aligned}$$

To get an essentially uniform alarm rate, choose $\varphi(\cdot)$ such that $\psi(\delta) = \alpha^*$ for all $\delta \in [0, 1)$; here $\alpha^* = -\log \alpha/n$, where α is a measure for the likelihood of false alarms (for instance 0.05). Unfortunately, $\varphi(\cdot)$ cannot be solved in closed form, but it can be obtained numerically (e.g. by bisection).

5 Numerical Evaluation

In this section we present the results of our numerical experimentation and focus on testing the procedures proposed for M/G/ ∞ in Section 4.2 (for which we could rely on a fairly explicit characterization of the threshold function $\varphi(\cdot)$). We picked the load parameter value based on the findings reported in [15] (where real VoIP data traces originating from one of the Italian service providers were analyzed). Several different types of numerical experiments validating proposed methods can be found in [8].

We then consider the situation that Y_1 up to Y_{100} are samples of the number of customers in an M/M/ ∞ queue, at epochs $\Delta, 2\Delta, \dots, 100\Delta$; Y_0 is sampled according to the equilibrium distribution (1). For these first 100 observations we chose $\lambda = 320$ and $\mu = 1$, leading to $\varrho = 320$. Then Y_{101} up to Y_{200} are generated in an analogous way, but now with $\lambda = 375$ (so that $\bar{\varrho} = 375$). Assuming a

maximum allowable blocking probability of 0.1%, the value $\bar{\rho} = 375$ corresponds with $C = 423$ lines. It is easily verified that choosing $\Delta = 10$ makes sure that $|U_{k,\ell}| \cdot \mathbb{P}(B^* > t) < \varepsilon_{\max}$, for an ε_{\max} of 0.01, using the procedures developed in Section 4.1.

We take windows of length 50, that is, we test whether H_0 should be rejected based on data points Y_i, \dots, Y_{i+49} , for $i = 1$ up to 151. The first window in which the influence of $\bar{\rho}$ is noticeable is therefore window 52. 500 runs are performed and two sampling regimes are considered, i.e., $\Delta = 1$ and $\Delta = 60$ (which could correspond to 1 second and 1 minute sampling in the real system). In line with what we mentioned earlier, for $\varepsilon_{\max} = 0.01$ the value of $\Delta = 1$ would be too low to ensure that the dependence between subsequent samples is sufficiently low. Nevertheless, such an experiment is interesting as it allows to assess the performance of the proposed methods if some assumptions are not fulfilled. Note that from a practical standpoint, the communication system administrator may be actually interested in sampling at a relatively high frequency as it allows for a very quick reaction (orders of seconds rather than minutes in our example) to the developing overload in the network, but this is evidently possibly at the expense of a higher false alarm ratio.

Figs. 1 and 3 show the detection ratio as a function of the window id. In the frequent sampling regime ($\Delta = 1$), due to the higher correlation of the samples, we observe a somewhat higher false alarm ratio (12%) than desired (5%). This is not the case for $\Delta = 60$, for which the observed false alarm ratio is around just 3%.

Clearly, from window 101 on all observations have been affected by the load change. For window i between 52 and 101, one could (within the window that consists of 50 observations) detect a load change at the earliest at the $(101 - i)$ -th observation — this is what could be called the ‘true changepoint’; in addition, we call the ratio of $101 - i$ and the window length 50, which is a number between 0 and 1, the ‘true delta’, in line with the meaning of δ in Section 4.2. Figs. 2 and 4 provide insight into the spread of the location of the detected changepoint position within the detection window, i.e., δ^* . It shows that the dispersion of the detected changepoint around the ‘true changepoint’ is low, indicating that the changepoint, once it occurs, is localized with sufficient precision. Again, better results are obtained if the correlation between the observations is reduced, which is typically the case for larger Δ .

Finally, Figs. 5 and 6 show the empirical cumulative distribution function of the number of customers present in the system at the moment an alarm is issued. More frequent sampling ($\Delta = 1$) allows for a somewhat lower mean (viz. 359) and median (359 as well) values, as compared to the ‘sparse sampling mode’ (i.e., $\Delta = 60$, leading to a mean of 370.2 and a median of 370), while the standard deviation is in both cases similar (21.6 and 21.3, respectively). Obviously, one would prefer lower values here, as it would allow for detecting the changepoint *before* arriving at the new equilibrium.

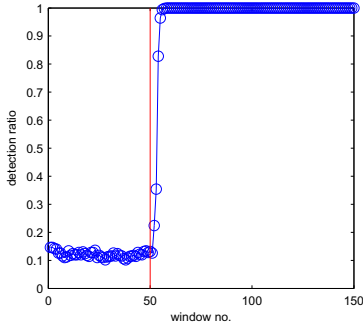


Fig. 1. Detection ratio, $\Delta = 1$

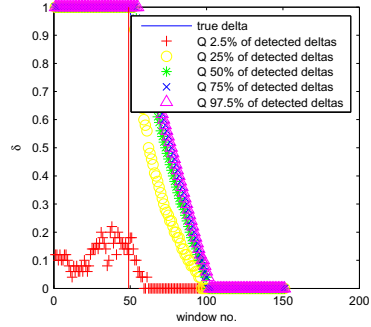


Fig. 2. Detection epoch, $\Delta = 1$

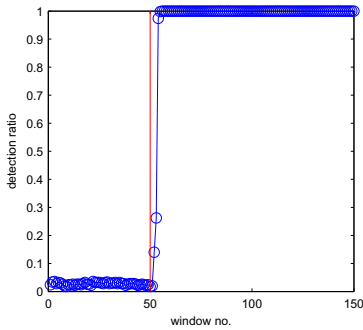


Fig. 3. Detection ratio, $\Delta = 60$

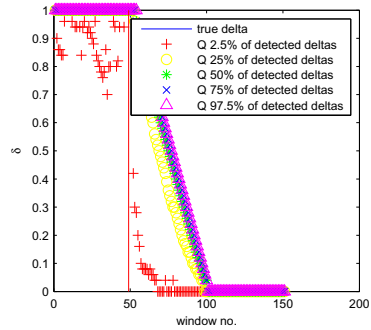


Fig. 4. Detection epoch, $\Delta = 60$

6 Results Using Real VoIP Data

In this section we present results obtained by applying our anomaly detection method to a real data trace. First of all, one should realize that a direct application of the proposed methods to detecting change points in VoIP data is not obvious, due to the fact that we assume that before a change there was a period in which the value of ϱ was constant (the ‘baseline model’, stationarity assumption). Throughout a day this is usually not the case (a typical one-working-day record of the number of calls is shown on Fig. 7).

One can remedy this effect by (i) either selecting time windows during which traffic is roughly constant (for instance the ‘plateau’ before lunch), and then directly apply the above methodology, (ii) or incorporating the changing ‘baseline’ (or: trend) into the detection mechanism. In this chapter we elaborate on the former approach; for the sake of completeness we briefly describe the latter approach now. It essentially relies on the fact that under the mild condition of ‘locally-Poisson’ call arrivals, and recalling that for large loads the Poisson-distributed

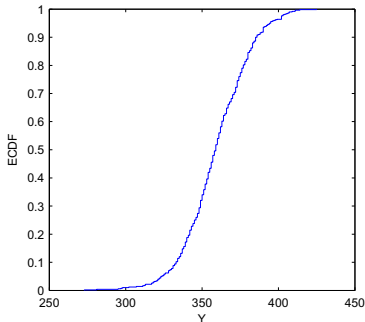


Fig. 5. Number of customers in the system at the detection time, $\Delta = 1$

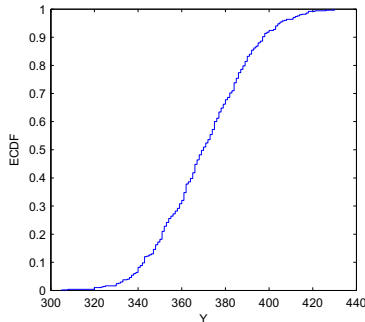


Fig. 6. Number of customers in the system at the detection time, $\Delta = 60$

number of calls can be approximated by an appropriate Normal random variable, we can set up a trend removal procedure that yields (roughly) standard Normal residuals. From that point on, we are interested in detecting a changepoint in a Gaussian sequence (rather than a Poissonian sequence). In [9] it is pointed out how to develop a test similar to the one described in Section 4.

To illustrate the performance of approach (i) mentioned above, we select one day from our repository, and describe the outcome of the tests. Nights were removed from the original dataset (as the number of calls is very low) and the curve ‘pattern’ is a moving average taking into consideration the observations from the past 5 weeks at the given time moment. Figure 7 should be interpreted as follows.

- On the *left scale* we record the actual and average number of calls (‘pattern’) based on observations from five previous weeks.
- On the *right scale* we record the relative position of the anomaly detected in the window of 50 samples which ends at the given time point (meaning that we have decided to skip the first 49 values as they would require readings from a previous day). A value of 1 means ‘no anomaly detected’, while for example a value of 0.94 observed at time point $t = 926$ means that at that moment the system reports an anomaly, and declares it has happened 3 observations before ($t = 923$) (as for a detection window of size 50, the distance between two consecutive observations is 0.02 and $1 - 0.94 = 0.06 = 3 \cdot 0.02$). Later on, we again observe a series of readings with no alarm reported. Then, at the onset of the ‘afternoon peak’ ($t = 955$), after some uncertainty at the beginning, we observe a consistent period that the detector reports that the number of calls is significantly higher than average, which is confirmed by a visual inspection. From $t = 1032$ on, the system again declares no anomaly.

Observe, that the proposed method is capable of not only detecting an overload, defined as a situation when the system approaches its capacity limits, but also the situation that the number of calls, while still being below the aforementioned

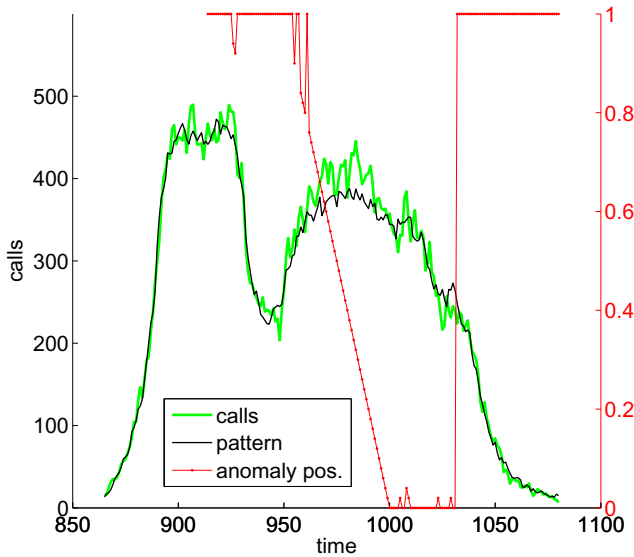


Fig. 7. Real data example

capacity limits, grows (falls) faster (slower) than the trend. Such information can be useful for example in call centers, as it may indicate the need for more staff than was initially planned.

7 Concluding Remarks and Discussion

In this chapter we presented procedures that are capable of detecting load changes, in a setting in which each connection consumes roughly the same amount of bandwidth (with VoIP as the leading example). We designed testing procedures, relying on large-deviations theory. In passing, we found results for the transient of the $M/M/\infty$ and $M/G/\infty$ systems, which are of independent interest. Simulation experiments for one of the proposed methods demonstrate its adequate capability of tracking load changes. Finally, we have demonstrated the applicability of the considered methods in a setting with real VoIP data.

References

1. Blanc, J., van Doorn, E.: Relaxation times for queueing systems. In: de Bakker, J., Hazewinkel, M., Lenstra, J.K. (eds.) *Mathematics and Computer Science. CWI Monograph*, vol. 1, pp. 139–162. North-Holland, Amsterdam (1984)
2. Bucklew, J.: *Large Deviation Techniques in Decision, Simulation, and Estimation*. Wiley, New York (1990)

3. Ho, L., Cavuto, D., Papavassiliou, S., Zawadzki, A.: Adaptive/automated detection of service anomalies in transaction-oriented WANS: Network analysis, algorithms, implementation, and deployment. *IEEE Journal of Selected Areas in Communications* 18, 744–757 (2000)
4. Mandjes, M.: *Large Deviations of Gaussian Queues*. Wiley, Chichester (2007)
5. Mandjes, M., Saniee, I., Stolyar, A.: Load characterization, overload prediction, and load anomaly detection for voice over IP traffic. *IEEE Transactions on Neural Networks* 16, 1019–1028 (2005)
6. van de Meent, R., Mandjes, M., Pras, A.: Gaussian traffic everywhere? In: *Proc. 2006 IEEE International Conference on Communications, Istanbul, Turkey* (2006)
7. Münz, G., Carle, G.: Application of forecasting techniques and control charts for traffic anomaly detection. In: *Proc. 19th ITC Specialist Seminar on Network Usage and Traffic, Berlin, Germany* (2008)
8. Mandjes, M., Żuraniewski, P.: $M/G/\infty$ transience, and its applications to overload detection. *Perf. Eval.* 68(6), 507–527 (2011)
9. Mata, F., Żuraniewski, P., Mandjes, M., Mellia, M.: Anomaly Detection in VoIP Traffic with Trends. In: *Proc. of ITC 2012, Kraków, Poland* (2012)
10. Shwartz, A., Weiss, A.: *Large Deviations for Performance Analysis*. Chapman and Hall, London (1995)
11. Siegmund, D.: *Sequential Analysis*. Springer, Berlin (1985)
12. Tartakovsky, A., Veeravalli, V.: Changepoint detection in multichannel and distributed systems with applications. In: *Applications of Sequential Methodologies*, pp. 331–363. Marcel Dekker, New York (2004)
13. Thottan, M., Ji, C.: Proactive anomaly detection using distributed intelligent agents. *IEEE Network* 12, 21–27 (1998)
14. Żuraniewski, P., Rincón, D.: Wavelet transforms and change-point detection algorithms for tracking network traffic fractality. In: *Proc. NGI 2006, Valencia, Spain*, pp. 216–223 (2006)
15. Żuraniewski, P., Mandjes, M., Mellia, M.: Empirical assessment of VoIP overload detection tests. In: *Proc. NGI 2010, Paris, France*, pp. 1–8 (2010)

Distribution-Based Anomaly Detection in Network Traffic

Angelo Coluccia¹, Alessandro D’Alconzo², and Fabio Ricciato^{1,2}

¹ University of Salento, Lecce, Italy

² Forschungszentrum Telekommunikation Wien, Vienna, Austria

Abstract. In this Chapter we address the problem of detecting “anomalies” in the global network traffic produced by a large population of end-users. Empirical distributions across users are considered for several traffic variables at different timescales, and the goal is to identify statistically-significant deviations from the past behavior. This problem is casted into the framework of hypothesis testing. We first address the methodology for dynamically identifying a reference for the null hypothesis (“normal” traffic) that takes into account the typical non-stationarity of real traffic in volume and composition. Then, we illustrate two general distribution-based detection approaches based on both heuristic and formal methods. We discuss also operational criteria for dynamically tuning the detector, so as to track the physiological variation of traffic profiles and number of active users. The Chapter includes a final evaluation based on the analysis of a dataset from an operational 3G network, so as to show in practice the detection of real-world traffic anomalies.

1 Introduction

Modern wide-area networks are managed by network operators that offer connectivity to their customers through an access infrastructure. This can be wired — mostly xDSL lines and, to a lesser extent, fiber lines (FTTH) — or wireless, like e.g. third-generation (3G) cellular networks and the upcoming 4G Long-Term Evolution (LTE) [18]. In this Chapter we address the problem of AD in the network traffic produced by a large population of end-users. We focus on *large-scale* anomalies, i.e. events that affect many users at the same time, which are of particular interest from the network operator’s viewpoint.

The approach we discuss is based on traffic distributions. Most AD tools consider traffic flows, namely based on TCP/IP tuples or other IP-based identifiers — e.g., Origin-Destination (OD) pairs [13] or port/address tuples [7]. When traffic can be feasibly mapped to users, it is possible to adopt a different “view” of the network based on network-wide distributions *across users*. The prerequisite of this approach is that traffic variables can be univocally associated to individual users. Such an association can be easily obtained in access networks from physical line and/or customer specific identifiers (e.g. IMSI in mobile networks), avoiding the ambiguities of IP addresses (dynamic assignment, spoofing). In other cases

such as e.g. backbone networks the association might be more difficult to obtain in practice.

Whenever the association above can be obtained, it is possible to look at a more sophisticated approach instead of the classical scalar-based approach. The great advantage of the former compared to the latter is that considering the *entire distribution across users* allows to detect also macro-events that do not cause appreciable changes in the total traffic volume, as shown later in §5.3 on a real dataset.

AD in operational contexts requires to define not only a detection rule but also a *reference identification algorithm* to learn the “normal” traffic profile. Furthermore, the whole scheme should be adaptive in order to cope with the typical characteristics of network traffic, such as time-of-day effect, trends and physiological variations due to several root causes — we will address these issues in §3. The challenge is that one would need an analytical model parsimonious enough to be mathematically tractable via statistical processing techniques, but at the same time versatile enough to fit different traffic variables, at different timescales and aggregation levels.

As a matter of fact, in operational contexts heuristic methods are often preferred to more theoretical approaches in order to obtain low-complexity “quick-and-dirty” rules designed upon the actual characteristics of the real traffic. However, these schemes are often hardly applicable to different settings. In general, the whole design of an AD tool should be driven by the operational requirements of *versatility*, *adaptiveness* and *low-complexity*. Versatility is required in order to statistically model different traffic variables at different timescales and aggregation. Moreover, the detection rule should adapt to changes in the network architecture and traffic composition with only minor parameter tuning. Finally, handling on-line a sufficiently large number of variables and users requires that the whole scheme has low complexity.

The rest of this Chapter is organized as follows. In §2 we formalize the AD problem within the framework of hypothesis testing, discussing also the definition of proper divergence metrics for distributions. In §3 we address the identification of a suitable reference set for the “normal” traffic, while in §4 we present two general approaches to distribution-based AD. In §5 we discuss some fundamental issues that arise when moving to operational contexts, and illustrate results obtained by applying the distribution-based approach to a dataset from an operational network. We summarized the Chapter in §6.

2 Anomaly Detection Framework

2.1 Problem Formulation

We consider a generic network (wireless or wired) serving a large population of users, where a passive *monitoring system* is able to associate each individual packet to the end-user that sent or received it. Such systems are usually deployed in operational networks for management and troubleshooting. For each user a set

of counters associated to several traffic variables is thus available, e.g. the “number of TCP SYN packets sent in uplink to port 80” or the “number of distinct IP addresses contacted”. Counters are aggregated at different time granularity (from minutes up to hours) to enable multi-scale analysis. Each variable is analyzed independently from the others, therefore the system can be considered as an array of parallel processing modules, each working on a temporal *series* of univariate samples. The empirical distributions of sample values are usually binned, often logarithmically to account for heavy-tails and wide range span, and represent the input data to the AD system.

The detection task is to test whether the current distribution $p_{\text{test}}(t_k)$ is consistent with the reference (null hypothesis \mathcal{H}_0) or should be considered anomalous (hypothesis \mathcal{H}_1). For the null hypothesis \mathcal{H}_0 at time bin k , a suitable *reference identification algorithm* must identify a set $\mathcal{S}_0(k|k-1, k-2, \dots)$ of past distributions representative of the “normal” behavior (more comments on this point below). Omitting the temporal dependency, we write:

$$\mathcal{H}_0 : p_{\text{test}} \in \mathcal{S}_0 \quad \text{vs.} \quad \mathcal{H}_1 : p_{\text{test}} \notin \mathcal{S}_0 \quad (1)$$

2.2 Divergence Metrics

The choice of an appropriate divergence metric is a key building block of the distribution-based approach, important both for the reference identification and detection tasks. The general form of a divergence metric is represented by the class of *f-divergences*, which for continuous distribution p and q is defined as:

$$D_f(p||q) = \int_{\Omega} f\left(\frac{dp}{dq}\right) dq$$

For data samples defined over a discrete probability space Ω , p and q are the probability mass functions (pmf). A simple example of f-divergence is the *total variation distance*, defined as:

$$\delta(p, q) = \frac{1}{2} \sum_{\omega \in \Omega} |p(\omega) - q(\omega)|$$

The class of information-theoretic distance measures (f-divergences, also called Ali-Silvey) have various geometric invariance properties that are of interest for applications [1,2,14]. Besides the total variation distance, the most important examples are *Hellinger distance* and the *Kullback-Leibler* (KL) divergence, or *relative entropy*. The former is defined as:

$$H(p, q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{\omega \in \Omega} (\sqrt{p(\omega)} - \sqrt{q(\omega)})^2}$$

while the latter, which is the most common divergence used to measure the difference between two distributions, is given by [19]:

$$D(p||q) = \mathbb{E}_p \left[\log \frac{p(\omega)}{q(\omega)} \right] = \sum_{\omega \in \Omega} p(\omega) \log \frac{p(\omega)}{q(\omega)} \quad (2)$$

where the sum is taken over the atoms of the event space Ω , and by convention (following continuity arguments) $0 \log \frac{0}{q} = 0$ and $p \log \frac{p}{0} = \infty$.

The KL divergence provides a non-negative measure of the statistical divergence between p and q . It is zero if and only if $p = q$, and for each $\omega \in \Omega$ it weights the discrepancies between p and q by the probability p . The KL divergence has several optimality properties that make it ideal for representing the difference between distributions. It contains the so called *likelihood ratio* p/q , whose importance derives from the Neyman-Pearson theorem [19,6]. In fact, it can be shown that in a hypothesis testing scenario, where a sample must be classified as extracted from p or q , the probability of misclassification is proportional to $2^{-D(p||q)}$ (Stein's lemma [19, §7]). Note that the KL divergence is not a distance metric, since it is not symmetric and does not satisfy the triangular inequality.

Building upon the KL divergence, in [4] a more elaborated metric is given:

$$L(p, q) = \frac{1}{2} \left(\frac{D(p||q)}{H_p} + \frac{D(q||p)}{H_q} \right) \quad (3)$$

where $D(p||q)$, $D(q||p)$ are defined accordingly to eq. (2), while H_p and H_q are the entropy of p and q , respectively, i.e.:

$$H_p = - \sum_{\omega \in \Omega} p(\omega) \log p(\omega)$$

and analogously for H_q .

The rationale for dividing the KL divergence $D(p||q)$ by the entropy — an approach previously used by Khayam *et al.* in [12] in the field of wireless channel modeling — is based on an information-theoretic interpretation. In fact, when the base-2 logarithm is used in eq. (2), $D(p||q)$ gives the average number of additional bits (overhead) needed to encode a source p with a code optimal for q . This is the *absolute* overhead (in bits) caused by replacing p with q . Since H_p represents the average number of bits required to encode p , the ratio $\frac{D(p||q)}{H_p}$ represents the *relative* overhead. Therefore, the result is a relative divergence metric. Moreover, the lack of symmetry of the KL divergence can be inconvenient in certain scenarios, particularly in presence of events with very low probability in only one of the two distributions — in which case $D(p||q)$ and $D(q||p)$ can take very different values (see e.g. the example in [22]). Although some different proposals have been made to overcome this limitation [11], the simplest strategy is to average the two values as in eq. (3).

3 Reference Identification

The dynamic identification of a set of distributions representative of the “normal” traffic, which constitutes the reference for the \mathcal{H}_0 hypothesis, must take into account the intrinsic peculiarities of real traffic. Several studies have been

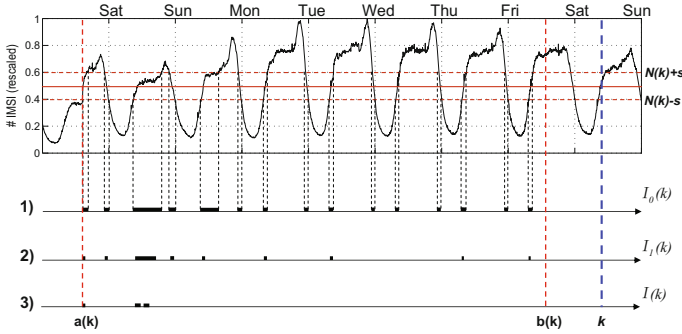


Fig. 1. Algorithm for the dynamic identification of the reference set

devoted to the analysis of the traffic structural characteristics from network measurements in different contexts [10,8,3,5,15] which all recognize a marked *non-stationarity* in both aggregate volumes and statistical distributions. In summary, the following structural characteristics can be identified:

- the traffic is *non-stationary* due to time-of-day variations;
- *steep variations* occur at certain hours, particularly in the morning and in the evening;
- many traffic variables exhibit strong 24-hours pseudo-periodicity;
- some variables show marked differences between working days and weekends/festivities.

We remark that such variations do not only apply to the total traffic volume and number of active users, but also to the entire *distribution* of each traffic variable. Distribution changes are due to variations in the traffic composition — e.g., relative share of active terminal types (handsets vs. laptops) and application mix (see e.g. [9]) — and in the individual user behavior (e.g., human-attended sessions are longer at evening and during the weekends).

To operate in a real environment, the identification of a reference to represent the “normal” behaviour — which is ancillary to the detection of the anomalies — must be able to cope with such characteristics, and should be versatile enough to be applicable to any traffic variable at different timescales. Indeed, traffic variables do share the same structural ingredients, but combined in many different ways: for instance, daily/weekly pseudo-cycles are typically strong in traffic variables linked to human-attended applications, while they may be much weaker in machine-to-machine traffic. Therefore, only a reference identification algorithm that does not rely upon specific traffic characteristic can be applied without modifications to traffic variable with very different structural characteristics and at different timescales, regardless to the distribution shape.

The classical and simplest solution is to use a sliding window approach, by considering N_w past timebins, i.e. $k-1, k-2, \dots, k-N_w$ if k is the index corresponding to the current timebin. The underlying idea is that the most recent observations carry the most correlated information about the traffic process,

hence can be used to build a reference of the normal behavior against which the timebin under scrutiny is compared. However, this simple approach does not take into account all the characteristics of the traffic, in particular the presence of physiological changes in the traffic at certain hours of the day that do not correspond to anomalies. Moreover, it does not exploit an important information which is present into the traffic process, i.e. the pseudo-periodicity: in fact, if the same behavior (e.g. an abrupt change) repeats systematically over time, then it may be not interpreted as an anomaly.

The single-window approach discussed above can be extended by considering a dual-window approach. In particular, the idea is to include in the sliding window also N'_w timebins corresponding at the same time in past days. In this way the considered indexes are $k - 1 - hN_{tbpd}, k - 2 - hN_{tbpd}, \dots, k - N_w - hN_{tbpd}$ for $h = 0, 1, \dots, N'_w$, where N_{tbpd} is the number of timebins per day (e.g., $N_{tbpd} = 24$ for 1-hour timescale). The drawback of this solution is that many traffic variables exhibit different statistical characteristics in weekend/festivity days compared to working days. Therefore, the reference set is usually inflated with too heterogeneous timebins.

A possible enhancement of the dual-window approach has been proposed in [4]. As depicted in Fig. 1, given a new sample at time k of size $N(k)$ — i.e., the number of active users with non-zero counter for the traffic variable at hand — in the first step the reference identification algorithm picks the subset $\mathcal{I}_0(k)$ of past time bins with samples of similar size, formally $\mathcal{I}_0(k) = \{j | N(k) - s \leq N(j) < N(k) + s\}$. Such size-based criterion avoids comparing samples with very different statistical significance — note that the number of active users can vary across two orders of magnitude during the 24 hours. Only the most recent observations up to a maximum age (e.g. last four weeks) are selected: this ensures that the reference identification process tracks the slow evolution trend of normal traffic. In a second refinement step, the subset of elements in $\mathcal{I}_0(k)$ with the smallest divergence from the current observation is picked: in this way samples related to different usage profiles — e.g., different time-of-day and/or type of day (working day vs. weekends/festivities) — are filtered out. The residual set $\mathcal{I}_1(k)$ might still contain residual heterogeneous samples. To eliminate these, in the third step an heuristic graph-based pruning procedure identifies the dominant subset with highest coherence: samples are mapped to nodes, with edges weighted proportionally to the KL divergence among them. The algorithm divides the nodes in two sets so as to maximize their mutual distance, and finally the larger one is picked as the final reference set $\mathcal{I}(k) \equiv \mathcal{S}_0(k | k - 1, k - 2, \dots)$. The overall procedure is designed to maximize coherence within the reference set, so as to preserve good sensitivity of the detection process. It should be remarked that past observations (samples) that were previously marked as “anomalous” by the detector are excluded from the reference identification procedure, in other words only samples marked as “normal” are taken as candidates. This introduces a sort of feedback loop, as the output of the detector for past samples impacts the identification of the reference and therefore influences the future decisions. For further details the interested reader is referred to [4].

In the following we assume that a reference identification algorithm provides at any instant k the reference set $\mathcal{S}_0(k|k-1, k-2, \dots)$ as input to the detector.

4 Distribution-Based Detectors

To translate (1) into a computable test, i.e. a *detector*, it is necessary to map distributions to a set of parameters that are representative of their characteristics. This is necessary since the hypothesis test requires a precise definition of the parameter set, which are tested in order to assess which of the two hypothesis (\mathcal{H}_0 non-anomalous, \mathcal{H}_1 anomalous) is statistically more likely. The identification of the characteristic parameters is not a trivial task, and reminds the issue of *machine learning* about feature selection. It should be done in a way that preserves the statistical properties more informative for the AD task, without including parameters that are non-informative, i.e. represent a noise that can “confuse” the detector. Either heuristic-based or more theoretically-founded approaches can be used. Both options are addressed in the following.

4.1 Heuristic-Based Detection

In the heuristic approach the comparison between the current distribution $X(k)$ and the associated reference set $\mathcal{S}_0(k|k-1, k-2, \dots)$ involves in general the computation of two compound metrics based on the divergence $L(\cdot, \cdot)$ (ref. eq. (3)). The first one, which can be referred to as *internal dispersion* $\Phi_\alpha(k)$, is a synthetic indicator characterizing the elements in the reference set. It has the role of defining an acceptance region for the test, and is typically extracted from the set of divergences computed between all the pairs of distributions in the reference set, e.g. the α -percentile.

The parameter α must be tuned so as to adjust the sensitivity of the detection algorithm, a point covered later in §5: it defines the maximum size of the distribution deviation that can be accounted to “normal” statistical fluctuations. In other words, it determines the size of the detectable events and therefore the false alarm rate.

Similarly, it can be defined the *external dispersion* $\Gamma(k)$ as a synthetic indicator extracted from the set of divergences between the current distribution $X(k)$ and those in the reference set, e.g. the mean.

The detection scheme is based on the comparison between the internal and external metrics. If $\Gamma(k) \leq \Phi_\alpha(k)$ then the observation $X(k)$ is marked as “normal”, i.e. the hypothesis \mathcal{H}_1 is rejected.

Besides the general idea sketched above, a number of issues must be addressed to avoid that the reference set becomes outdated and/or is inflated by anomalous timebins. A possible solution has been proposed in [4]. There, when the detector decides for \mathcal{H}_0 , the boundaries of the observation window are updated by a simple shift, i.e. $a(k+1) = a(k) + 1$ and $b(k+1) = b(k) + 1$. Conversely, the violation condition $\Gamma(k) > \Phi_\alpha(k)$ triggers an alarm, and $X(k)$ is marked as “anomalous”. The corresponding timebin k is then included in the set of anomalous timebins

$\mathcal{M}(k)$ and will be excluded from all future reference sets. In this case only the upper bound of the observation window is shifted, while the lower bound is kept to the current value, i.e. $a(k+1) = a(k)$ and $b(k+1) = b(k) + 1$. This update rule is meant to prevent the reference set from shrinking in case of persistent anomalies. In fact, only the timebins in $\mathcal{W}(k) \setminus \mathcal{M}(k)$ are considered for the reference set, where $\mathcal{W}(k)$ is the set of all timebins in the observation window.

Notably, the recomputation of $\Phi_\alpha(k)$ — as soon as the reference set gets updated — avoids that the decision about the nature of the current sample is done by comparing $\Gamma(k)$ with a static and predefined threshold, which choice injects always a certain degree of arbitrariness into the analysis. Conversely, the distributions in the reference set are used for dynamically learning an acceptability region for the “normal” traffic. This approach brings robustness to the detection method as it allows to cope with the marked non-stationarity of real traffic.

The steps performed by the algorithm are summarized in the pseudo-code of Fig. 2. Note that in the initialization phase, the observation window $\mathcal{W}(k)$ is obtained by setting the initial values of $a(k)$ and $b(k)$, which determine an initial window length of $l - r + 1$, and by excluding the timebins already in the anomalous timebin set $\mathcal{M}(k)$. Notably, the initial elements of $\mathcal{M}(k)$ must be set by manual labeling of the initial data — unless the latter is completely anomaly-free, in which case $\mathcal{M}(k) = \emptyset$.

```

SET  $\alpha, l, r, \mathcal{M}(k)$ ;
INITIALIZE  $\mathcal{W}(k)$ :
     $a(k) = k - l, b(k) = k - r, \mathcal{W}(k) = [a(k), b(k)] \setminus \mathcal{M}(k)$ ;
START

1. OBTAIN  $X(k)$  and  $N(k)$ ;
2. SELECT  $\mathcal{I}(k)$  from the observation window  $\mathcal{W}(k)$  by running the reference set identification algorithm;
3. CALCULATE the dispersions  $\Gamma(k)$  and  $\Phi_\alpha(k)$ ;
4. IF  $\Gamma(k) > \Phi_\alpha(k)$ 
    rise ALARM;
    SET  $\mathcal{M}(k) = \mathcal{M}(k) \cup \{k\}$ ;
     $a(k+1) = a(k)$ ;
ELSE
     $a(k+1) = a(k) + 1$ ;
END IF
5.  $b(k+1) = b(k) + 1$ ;
6. increase  $k$  by one and go-back to step 1.

```

Fig. 2. Pseudo-code of an heuristic-based detector

4.2 GLRT-Based Detection

Several Machine Learning techniques can be adopted to identify a set of features that synthetically represent the distributions. Such an idea is similar to a model fitting, but with a variable number of parameters sufficient to capture all the

characteristics relevant for AD. In the following we present this general approach, without restricting to particular techniques but rather keeping the discussion the most general. We simply assume that a given modeling algorithm is somehow able to associate to each empirical distribution a vector of parameters $\boldsymbol{\lambda}$ in the feature space.

By running the modeling algorithm a set of L parameter vectors $\boldsymbol{\lambda}_{\text{ref}}^{(s)}$, $s = 1, \dots, L$ are obtained from \mathcal{S}_0 , which collectively describe the reference behavior. Therefore, the test (1) can be rewritten more explicitly as follows:

$$\mathcal{H}_0 : \boldsymbol{\lambda}_{\text{test}} \in \Lambda_0 \quad \text{vs} \quad \mathcal{H}_1 : \boldsymbol{\lambda}_{\text{test}} \notin \Lambda_0 \quad (4)$$

where $\Lambda_0 \stackrel{\text{def}}{=} \{\boldsymbol{\lambda}_{\text{ref}}^{(1)}, \dots, \boldsymbol{\lambda}_{\text{ref}}^{(L)}\}$ represents the reference behavior and $\boldsymbol{\lambda}_{\text{test}}$ refers to the model parameters for the data under test.

In the test (4) both hypotheses are *composite* and vectorial. Therefore, it is not possible to adopt optimal tools like Neyman-Pearson test to maximize the probability of *correct detection* (PD) for a desired probability of *false alarm* (PFA). In such cases it is customary to resort to a procedure called *Generalized Likelihood Ratio Test* (GLRT). The GLRT somehow extends the optimal Neyman-Pearson likelihood test to the case of composite hypotheses, by adopting the ML estimation for the unknown parameters [20].

The starting point is the ratio between the maximum of the likelihood function over the parameter space under \mathcal{H}_1 and \mathcal{H}_0 , respectively, called *Generalized Likelihood Ratio* (GLR) and denoted by \mathcal{L} :

$$\mathcal{L} = \frac{\max_{\boldsymbol{\lambda} \notin \Lambda_0} f_{\mathcal{H}_1}(\boldsymbol{\lambda})}{\max_{\boldsymbol{\lambda} \in \Lambda_0} f_{\mathcal{H}_0}(\boldsymbol{\lambda})} \quad (5)$$

The distributions under the two hypotheses ($f_{\mathcal{H}_0}$ and $f_{\mathcal{H}_1}$) are chosen from a suitable parametric family with sufficient degrees of freedom to capture the significant characteristics of the data at hand. The exact expression of \mathcal{L} depends on the shape of the modeling distributions involving the parameters $\boldsymbol{\lambda}$ under the two hypotheses. In any case, the parameters need to be estimated from the data in order to compute the maximum.

By comparing the GLR with a threshold η one can decide whether the hypothesis \mathcal{H}_1 must be accepted or rejected, obtaining the GLR Test (GLRT):

$$\mathcal{L} \begin{matrix} >_{\mathcal{H}_1} \\ <_{\mathcal{H}_0} \end{matrix} \eta \quad (6)$$

The detection threshold η is chosen adaptively according to the desired PFA (see next Section), and can be set by characterizing \mathcal{L} analytically or via Monte Carlo techniques.

5 Application to Operational Networks

A number of issues must be faced when moving to the application of anomaly detection techniques to real data. To operate in real contexts, in fact, the parameters of the detector need to be dynamically adjusted to track the “physiological”

variations of real traffic, i.e. to tune the detection sensitivity. The fundamental problem is that the “ground truth” is unknown in practice, as discussed below.

5.1 Fundamental Issues

For heuristic-based detection, the key parameter in the alarm condition $\Gamma(k) > \Phi_\alpha(k)$ is the value α of the percentile that defines the internal dispersion $\Phi_\alpha(k)$. The *size* of the deviation to be marked as anomalous (or “statistically significant”) — hence the *size* of the detectable events, but also the rate of false alarms — depends on α . Analogously, a criterion must be defined to set the threshold η in the GLRT approach. Moreover, the tuning of α, η must be adaptive since the test statistic depends on the distributions within the reference set, which varies with time (ref. Fig. 3).

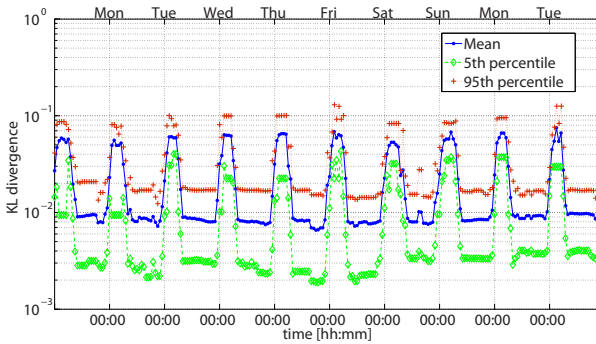


Fig. 3. KL divergence among distributions in the reference set (uplink SYN pkts)

In the radar field the presence/absence of a target are “hard facts”, on which it is possible to define a clear “ground truth” and hence a boolean true/false classification of the alarms. In network traffic, conversely, besides some kinds of “hard facts” that are certainly anomalous (e.g., a failure, a massive DDoS attack) there is a number of phenomena that evade the possibility of “hard classification”. These are typically events whose size spans a continuous range of values, from very small to very large. For example, consider a piece of malware that suddenly spreads to n mobile users, each of which immediately starts a scanning thread to propagate further the malware. If n is large, the infection can be clearly marked as a macroscopic anomalous phenomenon that the tool should report, and the operating staff should react upon. On the other hand, if n is very small, that is limited to only a handful of users, the infection is probably to be considered as a “physiological” phenomenon, not to be reported to the staff. But since n can take any intermediate value, from very small to very large, identifying the “right” border between what should be reported and what should be neglected is challenging. Clearly, any threshold setting on n — hence on α, η — would be arbitrary in this case. Down to an operational ground, instead of pursuing the “right” setting, we should seek for the most “useful” one.

Consequently, we must accept that the notion of (operational) utility involves a certain degree of subjective evaluation, which is unavoidable in this context.

These arguments have a serious impact on the *tuning* and *evaluation* of any statistical-based detection scheme for real traffic. Since the ground truth cannot be completely reduced to “hard facts”, and in any case it cannot be completely known (see e.g. discussion in [17,16]), the alarms generated by the system cannot all be classified as false/true, and the standard methods to assess the power and accuracy of the detector (e.g., ROC curves) become inapplicable.

5.2 Adaptive Threshold Setting at Run-Time

In practice, each alarm must be processed downstream by a human expert who must interpret it, and the cost (time, manpower) of such an interpretation is often heavy. Therefore, the penalty associated to false alarms, i.e. alarms caused by pure statistical fluctuations, is serious. Too many such alarms (alarm noise) would overload the human experts in charge of processing them, and ultimately undermine the credibility of the whole detection system among the experts. In other words, the tolerable level of false alarms is very low, hence the goal is to tune the sensitivity of the algorithm so as to obtain a very low PFA.

A possible solution is to approximatively characterize the test statistics $\Gamma(k)$ and \mathcal{L} , respectively for the heuristic-based and GLRT-based approaches, from the reference set itself — which by definition contains only \mathcal{H}_0 -type data. By taking one element $s \in \mathcal{S}_0$ as distribution under-test, and the remaining $L - 1$ as reference, it is possible to obtain a realization of the random variable $\Gamma(k)$ or \mathcal{L} . Therefore, by repeating the procedure for all elements in \mathcal{S}_0 we end up with an estimation of the test statistic distribution based on $L - 1$ samples. This method is known as “leave-one-out” [21]. Other solutions based on bootstrap are possible, but their computational cost may not be suited for on-line implementation. Finally, the detection thresholds η and $\Phi_\alpha(k)$, respectively for the heuristic-based and GLRT-based approaches, are refined by taking a percentile. A typical trend over time is depicted in Fig. 4(a).

5.3 Examples from a Live 3G Network

In the following we illustrate some results obtained by applying the distribution-based approach on two weeks of traffic taken from an operational 3G network. The dataset contains three anomalies of different nature, labelled as event “A”, “B” and “C”. They correspond, respectively, to a pre-planned maintenance intervention with rebooting of some network elements, a temporary bottleneck due to a hardware failure and the famous worldwide Skype outage occurred in August 2007¹.

We consider two traffic variables, the “number of TCP SYN packets in uplink” at 1-hour timescale (Fig. 4(a)) and the “number of destination ports” (ref. Fig. 5(a)). The curves represents the trend over time of the detection threshold,

¹ http://heartbeat.skype.com/2007/08/what_happened_on_august_16.html

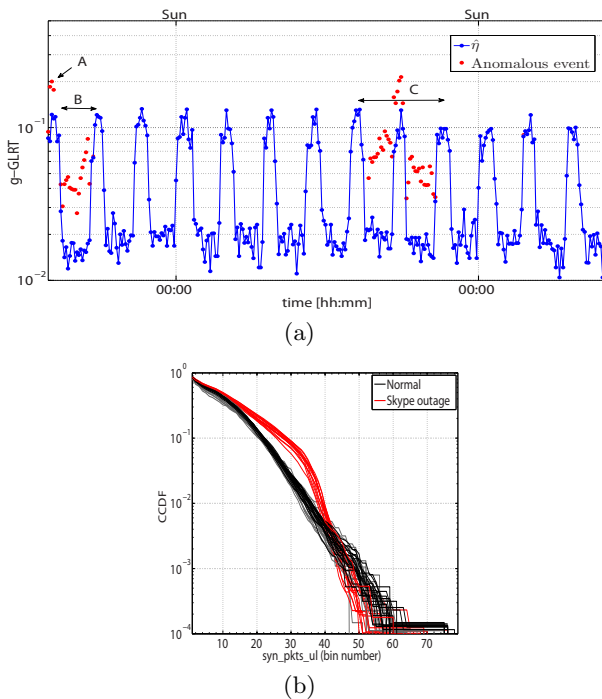


Fig. 4. Number of uplink SYN packets during the Skype outage: (a) time plot of detection threshold $\hat{\eta}$ and alarms (red circles) at 1-hour timescale; (b) empirical CCDFs.

whereas the circles mark the alarms, i.e. the points where the alarm condition was triggered. Note that the threshold raises at night, when the number of active users decreases considerably, hence statistical fluctuations become larger. In other words, the acceptance region dynamically follows the natural traffic variability thus avoiding false alarms at night.

In Fig. 4(a), from left to right, we see first a few isolated alarms occurring at night time (event “A”) due to a pre-planned maintenance intervention: immediately after the rebooting, clients re-open broken TCP connections, causing an excess of SYN packets. Note also that this event does not trigger any alarm in the other traffic variable “number of destination ports” (ref. Fig. 5(a)).

Following in Fig. 4(a), we observe a cluster of persistent alarms lasting an entire day (event “B”), due to a temporary bottleneck which was fixed during the following night: the affected mobile users reacted by re-starting slowed-down or stalled TCP connections, causing a distribution change in this variable that was correctly detected by our algorithm.

The third group of persistent alarms lasting for 48 hours is the Skype outage (event “C”), in both variables (Fig. 4(a) and Fig. 5(a)). The explanation of this anomaly follows. When a Skype client fails to connect to other (super)nodes, it probes for other hosts and port numbers to bypass possible firewalls. Due

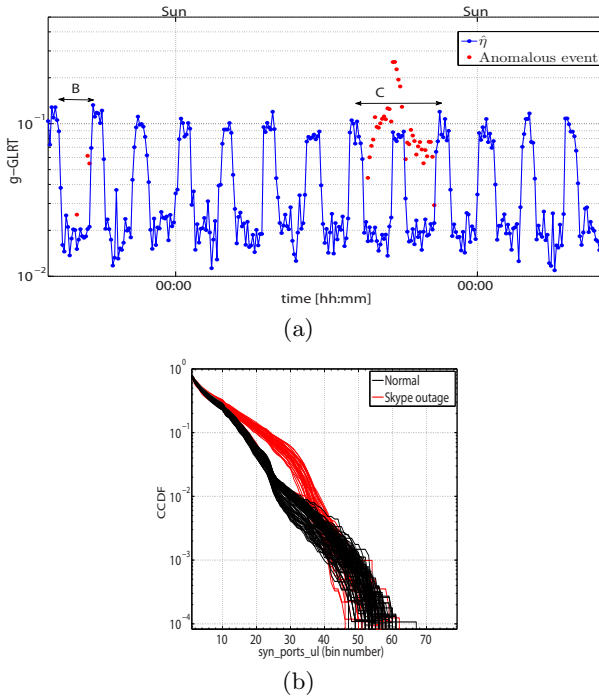


Fig. 5. Number of distinct destination ports of uplink SYN packets during the Skype outage: (a) time plot of the detection threshold and alarms (red circles) at 1-hour timescale; (b) empirical CCDFs.

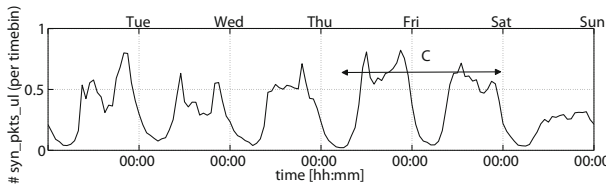


Fig. 6. Time diagram of the total number of SYN packets in uplink, same dataset of Fig. 5(a)

to the outage, the whole Skype network was temporarily down, and all active Skype clients in the monitored network reacted simultaneously by entering into probing mode. This caused a macroscopic change in the distribution across the user population of uplink SYN packets and contacted destination ports, as seen in the CCDF of Fig. 4(b) and Fig. 5(b) respectively². The AD system correctly raises a series of alarms for both such traffic variables during the entire duration

² Bins on the abscissa are identified merely by their order number since bin boundaries are subject to non-disclosure policy with the network operator.

of the anomaly. We remark that such event, which is easily revealed by looking at the entire distribution of SYN packets across mobile users, did not change appreciably the total number of SYN packets — the latter is reported in Fig. 6 (values are normalized due to non-disclosure policy). This observation confirms the superior detection power of distribution-based AD against classical methods based on the analysis of scalar time-series of total volume.

6 Conclusions

In this Chapter we have discussed an AD methodology based on distribution across users. We have first illustrated the characteristics of real traffic that must be taken into account in the design of a reference identification algorithm, presenting also a viable dynamic solution for this problem. Then, we have discussed two main approaches for distribution-based AD, one based on heuristic detection and one based on a formal GLRT. Both methods are general and do not require any *a priori* assumption about the actual data distribution, thus they are directly applicable to traffic variables with very different characteristics and at different timescales. We have also provided operational criteria of general applicability for the dynamic setting of the algorithm parameters, in particular the detection threshold. In the last part of the Chapter we have shown that the proposed methodology can effectively detect changes associated to real network problems, by analyzing a dataset from an operational 3G network.

References

1. Ali, S., Silvey, S.: A general class of coefficients of divergence of one distribution. *Journal of Royal Statistical Society* 28 (1966)
2. Csiszár, I.: Information-type measures of difference of probability distributions and indirect observations. *Studia Sci. Math. Hungar.* 2, 299–318 (1967)
3. Burgess, et al.: Measuring system normality. *ACM Transactions on Computer Systems* 20 (2002)
4. D’Alconzo, et al.: A distribution-based approach to anomaly detection for 3G mobile networks. In: *IEEE Globecom* (2009)
5. D’Alconzo, et al.: Distribution-based anomaly detection in 3G mobile networks: from theory to practice. *Int. J. of Network Management* (2010)
6. Dasu, et al.: An information-theoretic approach to detecting changes in multi-dimensional data streams. In: *INTERFACE 2006* (2006)
7. Gu, et al.: Detecting anomalies in network traffic using maximum entropy estimation. In: *IMC* (2005)
8. Lakhina, et al.: Structural analysis of network traffic flows. In: *ACM SIGMETRICS* (June 2004)
9. Svoboda, et al.: Composition of GPRS/UMTS traffic: snapshots from a live network. In: *IPS-MOME 2006* (2006)
10. Maier, G., Feldmann, A., Paxson, V., Allman, M.: On dominant characteristics of residential broadband internet traffic. In: *IEEE IMC* (2009)
11. Johnson, D.H., Sinanovic, S.: Symmetrizing the Kullback-Leibler distance. *IEEE Transactions on Information Theory* (March 2001)

12. Khayam, A., Radha, H.: Linear-complexity models for wireless MAC-to-MAC channels. *ACM Wireless Networks* 11 (2005)
13. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature. In: *ACM SIGCOMM* (2005)
14. Liese, F., Vajda, I.: *Convex statistical distances*. Teubner-Verlag (1987)
15. Ricciato, F., Coluccia, A., D'Alconzo, A., Veitch, D., Borgnat, P., Abry, P.: On the role of flows and sessions in internet traffic modeling: an explorative toy-model. In: *IEEE Globecom* (2009)
16. Ringberg, H., Roughan, M., Rexford, J.: The need for simulation in evaluating anomaly detectors. *ACM SIGCOMM Computer Communication Review* 38(1), 55–59 (2008)
17. Ringberg, H., Soule, A., Rexford, J.: Webclass: adding rigor to manual labeling of traffic anomalies. *ACM SIGCOMM Computer Communication Review* 38(1), 35–38 (2008)
18. Sesia, S., Toufik, I., Baker, M.: *LTE, The UMTS Long Term Evolution: From Theory to Practice*. J. Wiley & Sons (2009)
19. Thomas, J.A.T., Cover, T.M.: *Elements of Information Theory*. J. Wiley & Sons (1991)
20. Van Trees, H.L.: *Detection, Estimation, and Modulation Theory*. J. Wiley & Sons (2001)
21. Vapnik, V.N.: *Statistical Learning Theory*. J. Wiley & Sons (1998)
22. Song, X., et al.: Statistical change detection for multi-dimensional data. In: *13th ACM KDD 2007*, pp. 667–676. ACM (2007)

Part III
Quality of Experience

From Packets to People: Quality of Experience as a New Measurement Challenge

Raimund Schatz¹, Tobias Hofffeld², Lucjan Janowski³, and Sebastian Egger¹

¹ Telecommunications Research Center Vienna (FTW), Vienna, Austria
schatz,egger@ftw.at

² University of Würzburg, Institute of Computer Science, Germany
tobias.hossfeld@uni-wuerzburg.de

³ AGH University of Science and Technology, Krakow, Poland
janowski@kt.agh.edu.pl

Abstract. Over the course of the last decade, the concept of Quality of Experience (QoE) has gained strong momentum, both from an academic research and an industry perspective. Being linked very closely to the subjective perception of the end user, QoE is supposed to enable a broader, more holistic understanding of the qualitative performance of networked communication systems and thus to complement the traditional, more technology-centric Quality of Service (QoS) perspective.

The purpose of this chapter is twofold: firstly, it introduces the reader to QoE by discussing the origins and the evolution of the concept. Secondly, it provides an overview of the current state of the art of QoE research, with focus on work that particularly addresses QoE as a measurement challenge on the technology as well as on the end-user level. This is achieved by surveying the different streams of QoE research that have emerged in the context of Video, Voice and Web services with respect to the following aspects: fundamental relationships and perceptual principles, QoE assessment, modeling and monitoring.

1 Introduction

Understanding and measuring quality of communication services and underlying networks from an end-user perspective has attracted increased attention over the course of the last decade. This development has not only been driven by general trends such as the 'rise of the consumer' and the related emergence of the 'experience economy' [90]. In the context of networked communications it is mainly a consequence of increasing competition amongst stakeholders in the ICT, media and entertainment markets, the proliferation of resource intensive services (such as youtube.com) and the ever-present risk of customer churn caused by inadequate service quality. These trends create conflicting, challenging demands on the network operators and service providers involved: on the one hand, they need to develop and offer sophisticated high-performance infrastructures and services that enable high quality experiences that lead to customer satisfaction and loyalty [71, p.37]. On the other hand, they have to operate on a profitable basis in

order to remain successful in the long run. Parallel to these economic developments, we have been witnessing a growing awareness of the scientific community that technology-centric concepts like *Quality of Service* (QoS) are not powerful enough to cover every relevant performance aspect of a given application or service (cf. [35,100]) and understand the related value that people attribute to it as a consequence [71,10].

For these reasons, the concept of *Quality of Experience* (QoE) has gained strong interest, both from academic research and industry stakeholders. Being linked very closely to the subjective perception of the end user, QoE is supposed to enable a broader, more holistic understanding of impact and performance of network communication and content delivery systems and thus to complement (but not necessarily replace!) traditional perspectives on quality and performance.

This chapter provides an overview of recent QoE research and related challenges. To this end it first provides a general background to the concept. This is then followed by a discussion of the specific QoE issues and ongoing research for the three different service categories: voice communication, audio-visual multimedia and Web applications. By choosing an interdisciplinary point of view, this chapter aims to demonstrate that QoE not only represents a challenging field which brings together various scientific domains ranging from psychophysics to network monitoring. It also should show in which ways QoE helps us better understand multimedia communication systems and successfully improve the performance in ways the end user really needs and appreciates.

2 Background: Towards a New Understanding of Quality

This section provides an introduction to QoE by discussing its origins and evolution over time towards becoming a new paradigm for understanding quality. Furthermore, it provides an overview of subjective assessment methods, discusses generic relationships between Quality of Service (QoS) and QoE, and finally highlights various applications of the concept and related challenges.

2.1 Origins and Evolution of QoE

Understanding the origins of QoE requires a brief review of the recent history of communications quality assessment. During the 1990's, the notion of *Quality of Service* (QoS) has shaped the networked communications landscape to a substantial extent. As a consequence of having been repeatedly defined by various institutions (ISO, ITU-T, ETSI, IETF) in slightly different ways, the term 'QoS' can refer to the performance of networked services with three different meanings (cf. [36]):

1. The definition and assessment of service quality, class, and grade,
2. the specification of a contract between a customer and a service provider (i.e. SLAs),

3. QoS architectures of IP networks for controlling quality and improving performance (e.g. IntServ, DiffServ),

with only the first one being of actual relevance for our discussion of QoE¹

Interestingly, the ITU-T originally defined QoS as user-centric concept, namely as

"... the collective effect of service performance which determines the degree of satisfaction of a user of the service." [52]

In this respect, this definition clearly distinguishes between intrinsic, purely technical performance² and the performance perceived by the user. Later on, E.800 explicitly states that *"the essential aspects of the global evaluation of a service is the opinion of the users of the service. The result of this evaluation expresses the users' degrees of satisfaction."*, clearly emphasizing that end-user perception and the resulting opinion constitute an integral part of network and service quality definition and assessment.

However, contrary to this original definition, most QoS-related work actually focused on the investigation of purely technical, objectively measurable network and service performance factors such as delay, jitter, bitrate, packet loss – effectively reducing quality to a purely technology-centric perspective (cf. [100,17]). This *de facto* reduction of quality to network- and system-level performance parameters is also reflected in the QoS operationalizations that became dominant during that period, for example

"A set of quality requirements on the collective behaviors of one or more objects in order to define the required performance criteria." [60]

or the IETF's understanding of QoS as

"A set of service requirements to be met by the network while transporting a flow." [22]

In recognition of this gradual but remarkable reduction of scope, a counter-movement emerged in order to reintroduce user-centricity to quality assessment. Preceded by alternative terms like "subjective" or "user-perceived" QoS [40,13,9], the notion of *Quality of Experience* was introduced to the networking community by Moorsel [82] in the context of web-based services. In his paper, Moorsel argues for transcending mere performability measurement on a technical level by shifting the focus towards quality of experience and so called 'quality of business' metrics, thus also stressing the economic relevance of QoE. Using the example of an online shop, the paper introduces a framework that relates QoS with QoE and business-related 'QoBiz' metrics for each stakeholder in the communications ecosystem. Albeit falling short of an explicit QoE definition,

¹ A thorough survey of the concept of QoS can be found in ITU-T E.800 as well as [36].

² E.800 defines network performance as *"the ability of a network or network portion to provide the functions related to communications between users"* [52].

Moorsel clearly distinguishes QoE from QoS since QoE metrics "... *may have a subjective element to it ...*" while QoS measures not (i.e. perceived speed vs. throughput)[82]. The paper also emphasizes the multi-layered nature of quality (e.g. by distinguishing between system- and task-level QoS), a recurring theme in QoE research (cf. [110,59]).

Beyond web performance, the notion of QoE from this point on was rapidly adopted not only in the context of mobile communications (cf. [85,113]) but also in the domains of audio and video quality assessment (cf. [80,97,104,130], i.e. domains which already had a strong tradition of user centric quality assessment at that time. Indeed, each service type (voice, video, data services, etc.) tended to develop its own QoE community with its own research tradition. This has resulted in a number of parallel attempts to define QoE (as outlined in [100,101]), accompanied by an equally large number of QoE frameworks and taxonomies (see [72] for a comprehensive overview). Today, the definition presented by ITU-T Study Group 12 is still the most widely used (but also not fully agreed upon) formulation of QoE, defining the concept as "*The overall acceptability of an application or service, as perceived subjectively by the end user.*" Most importantly, this "*Includes the complete end-to-end system effects*" and "*May be influenced by user expectations and context.*" [56].

In this respect, ITU-T P.10 captures the essence of QoE by highlighting some of its main characteristics: subjective, user-centric, holistic, and multi-dimensional. Particularly as concerns the latter aspect, most frameworks and definitions found in the literature highlight the fact that QoE is determined by a number of hard and soft *influence factors* attributable either to the technical system, usage context or the user him/herself (see Fig. 1). This means that whether a user judges the quality of e.g. a mobile video service as good (or even excellent) not only depends on the performance of the technical system (including traditional network QoS as well as client and server performance)³, but to a large extent also on the context (task, location, urgency, etc.) as well as the user himself (expectations, personal background, etc.). The resulting level of complexity and breadth turns reliable and exact QoE assessment into a hard problem. Indeed, this is also one of the main reasons why as of today the scientific QoE community remains fragmented and has not agreed on a common QoE definition as well as a unified QoE framework yet.

As one of the most recent initiatives, the COST Action IC 1003⁴ has published a QoE definition whitepaper to further advance the required convergence process regarding this subject[17]. Version 1.1 of this whitepaper defines the QoE as "... *the degree of delight or annoyance of the user of an application or*

³ Note that the technical system generally comprises of a chain of components (sender, transmission network elements, receiver) that connect the service provider with the end-user. All these elements can influence technical QoS (and thus QoE) on different layers, predominantly in terms network- and application-level QoS.

⁴ The COST Action IC 1003 Qualinet known as 'European Network on Quality of Experience in Multimedia Systems and Services' has started in 2011 to coordinate research efforts in the field of QoE under one formal umbrella.

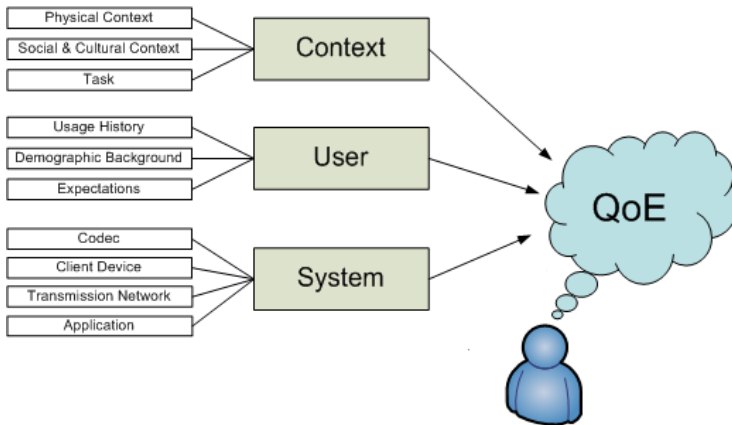


Fig. 1. QoE influence factors belonging to context, human user and the technical system itself

service. It results from the fulfillment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the users personality and current state". It thus advances the ITU-T definition by going beyond merely binary acceptability and by emphasizing the importance of both, pragmatic (utility) and hedonic (enjoyment) aspects of quality judgment formation⁵. In addition to QoE influence factors, the paper also highlights the importance of *QoE features*, i.e. recognized characteristics of the individual's experience that contribute to service quality perception. These features can be classified on four levels: direct perception (e.g. colour, sharpness, noisiness), interaction (e.g. responsiveness, conversation effectiveness), usage situation (e.g. accessibility, stability), and service (e.g. usability, usefulness, joy). These features can be represented in a multi-dimensional space and are not necessarily independent of each other. Within this framework, QoE can then be expressed as a vector of distances (either to the origin or to some ideal reference point) formed under the constraint of the aforementioned QoE influence factors[17].

2.2 QoE Assessment

Similar to QoS, the central question for QoE research and engineering is how to operationalize the concept in terms of performing reliable, valid, and objective measurements. This challenge is framed by the overarching question 'How can we quantify quality and how can we measure it?' Since inclusion of the human end-user's perspective is the defining aspect of QoE, conducting measurements merely on a technical level (e.g. by just assessing conventional end-to-end QoS integrity parameters) is not sufficient. In particular, QoE also accounts for user

⁵ The definitions of the terms used as well as further details can be found in the QoE definition whitepaper[17] itself.

requirements, expectations and contextual factors like task and location. Thus, quality assessment schemes are needed that act as translator between a set of technical (QoS) and non-technical (subjective and contextual) key influence factors and user perception, and ultimately, user experience. These can be categorized into subjective and objective quality assessment methods, depending on whether human subjects are involved in the assessment process or not.

Subjective Quality Assessment Methods are based on gathering information from human assessors (frequently referred to as 'test participants' or 'test subjects') who are exposed to different test conditions or stimuli during the process. In general, a panel of assessors is subjected to various quality levels (e.g. audio clips encoded using different settings) which leads to some form of explicit or implicit response. In most cases, quantitative methods derived from neighboring disciplines such as psychophysics and psychometrics are used to obtain information regarding assessors' judgment in the form of ratings that describe their perception of the respective quality experienced (cf. [79]). In addition, qualitative methods such as focus groups, interviews or open profiling [117] are used, particularly in order to find out which influence factors or features contribute to QoE and how [70]. Subjective tests are typically conducted in a controlled laboratory⁶ setting and require careful planning in terms of which variables and influence factors need to be controlled, measured and monitored. To this end, recommendations like ITU-R BT.500 and ITU-T P.910 provide detailed guidelines regarding choice of test conditions, rating scales, room setup as well as sequencing and timing of the presentation. The typical result of a subjective test campaign are the individual assessor's ratings which are typically aggregated into so-called mean opinion scores (MOS). The MOS expresses the average quality judgment of a panel regarding a certain test condition regarding the overall quality experienced or along a certain quality dimension (e.g. picture quality) [56]. It is typically based on an ordinal five-point scale: (1) bad; (2) poor; (3) fair; (4) good; (5) excellent. Note that most test designs rely on absolute scales (like the aforementioned five-point scale), but also relative Differential MOS (DMOS) or continuous methods are being used (see ITU-R BT.500 for details).

The MOS has become the de facto standard metric for QoE, a development that has become a subject of considerable debate (cf. [69,12,49]). Beyond the controversial use of ordinal grades for computing averaged scores, this debate is also nurtured by the fact that assessors' judgments are far from perfect [41] and are influenced by various user- and context-related parameters (including user assumptions and unconscious psychological factors) that are very hard to control or measure [15]. Therefore, a number of authors have proposed to complement subjective QoE ratings with alternative measures free from distortion by user opinion. Such *objective* QoE measures [69,15] can be task performance (e.g.

⁶ In addition to the lab, field trial methods for conducting studies under real-world conditions [81,105] as well as cost-effective crowdsourcing methods [18,45] have become popular in subjective QoE assessment.

quality and speed of goal completion) [69], physiological indicators (e.g. heart rate, skin conductance) [129,74] or user behavior in general (e.g. cancellation rates, viewing times) [65,25]. However, while such indirect quality indicators have shown to be more objective indeed, today they mainly have a complementary function since generalizable, conclusive mappings to QoE have not been found yet [79].

Objective Quality Assessment Methods. Today, subjective experiments are still the most accurate way to measure perceived quality and the only way to obtain reliable ground truths. However, they are also costly, time-consuming and very complex to implement due to the involvement of human end users. For the same reasons, subjective approaches cannot be applied to real-time in-service quality assessment. Therefore, objective QoE assessment methods (also called 'objective metrics') are being investigated with the purpose to automatically predict QoE at high accuracy on behalf of algorithmic processing of input parameters. However, objective metrics are only useful if their measurements closely correlate with subjective quality. Therefore, an integral part of the design process is the derivation of quality models that map quantifiable influence factors to predicted MOS values. To this end, the data obtained from subjective quality experiments (see above) is required to find model functions that provide an optimum fit with human quality perception (see also section 2.3).

According to [80] and [96], objective quality assessment approaches can be categorized on behalf of the following criteria:

1. Targeted service: Service type, e.g. IPTV, VoIP telephony, video conferencing, mobile TV, web browsing.
2. Model type: Utilization of a reference signal, i.e. Full Reference (FR), Reduced Reference (RR), No Reference (NR), as explained below.
3. Application: Codec testing, network planning, verification of Quality of Service classes, monitoring, etc.
4. Model input: Parametric description of the processing path (i.e. protocol information or planning values), additional payload information from bit-stream, reconstructed signal, combinations of parameters and signal, etc.
5. Model output: Overall quality or specific quality aspects in terms of MOS or another index.
6. Modeling approach: Psychophysical (i.e. explicit modeling of the human perceptual system, e.g. [131]) vs. empirical approaches (based on extracting characteristic system features by conducting experiments).

As regards model type, full reference (FR) metrics need both the original source and the transmitted signal of interest as depicted in Fig. 2 a). In contrast, no reference (NR) metrics estimate QoE on behalf of the output signal only while reduced reference (RR) refers to using only some information about the source signal. For service monitoring in real-time, reduced reference (RR) or no reference (NR) models are generally used, as they do not require costly acquisition and processing of the full reference signal. Therefore, they can be applied at different points in the network. For network planning purposes only NR models

can be used, because no signals are available during that phase [96]. In contrast, full reference (FR) models are typically used in laboratory settings where high accuracy is required and the reference signal is easily available. Based on this reasoning, three fundamental categories for instrumental quality models have emerged suitable for different assessment scenarios [79] as depicted in Fig. 2:

1. **Signal-based** models which assess the quality of signal (e.g. picture, sound), often by comparing with a reference (e.g. PESQ, ITU-T P.862.2).
2. **Parametric planning** models which predict quality just on behalf of planning parameters of the technical system (e.g. the ITU-T G.107 E-Model).
3. **Packet-level and bitstream** models which are typically used for monitoring purposes, often based on parameters that can be extracted from the transport stream and bitstream with little or no decoding (e.g. for quantifying the visual impact of packet-loss [63], existing standards are ITU-T SG 12, P.1201 and P.1202).

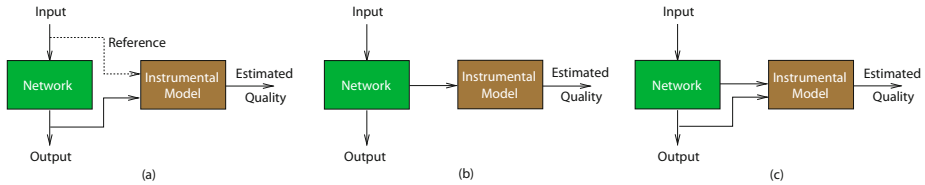


Fig. 2. Signal-based FR/NR (a), Parametric Planning (b) and Packet-level/bitstream Models (adapted from [77])

Despite obvious speed and economy advantages, the caveat with objective metrics is that they are only an approximation of a limited number of aspects of human quality perception. Therefore, they can provide inaccurate or inconclusive results, particularly when applied to new conditions they were not initially trained or designed for [79,70]. Objective metrics therefore need to be developed carefully, their application scope clearly defined and continuously validated against data from subjective experiments. For these reasons, much further research is required until QoE can be, if at all, accurately measured using objective metrics only.

2.3 Relationships between QoS and QoE

Albeit it has become evident that QoE is a multi-dimensional concept influenced by a number of system-, user- and context-related factors (see Section 2.1), it is also important to remember that in a majority of cases it is highly dependent on QoS. The main reason is that in any IP-based system, problems on traditional network QoS level (e.g. low available bandwidth, high delay, packet loss) but also on application-layer QoS (slow terminals and faulty operating systems) can have considerable (or even defining) impact on QoE (i.e. users annoyed by long waiting

times, noise, artifacts or outages). In addition, QoS belongs to the most business-relevant, actionable parameters for network and service providers [46]. For these reasons, the understanding and modeling of *generic relationships* between QoS and QoE has recently received considerable attention (cf. [32,47,21,99,109,26]).

In general, $QoE = \Phi(I_1, I_2, \dots, I_n)$ is a function of n influence factors I_j . In contrast, generic relationships focus on a single influence factor I in order to derive the fundamental relationship $QoE = f(I)$. Therefore, these relationships represent simple, unified and practicable formula expressing a mathematical dependency of QoE on network- or application-level QoS. They are thus applicable to online in-service QoE monitoring of QoS-related problems (e.g. as part of parametric planning or packet-layer models), enabling QoE management mechanisms that build on QoS monitoring [32]. Two prominent categories of such relationships which have been frequently observed in practice are logarithmic and exponential relationships.

Logarithmic Relationships – The Law of Weber-Fechner. A number of QoE experiments have identified relationships of the form $MOS = \alpha \cdot \log^{-\beta} \cdot QoS + \gamma$ between QoE and QoS, be it in the context of web browsing (cf. ITU-T G.1030 and [50]), file downloads [99] or VoIP services [122]. For example, the results of one VoIP quality study in [122] demonstrate that when using the Speex codec, the MOS as a function of the bitrate used can be well matched by logarithmic regressions.

Systematic studies of these observations [99,101] revealed that these logarithmic relationships can be explained on behalf of the well-known Weber-Fechner Law (WFL) [127], which in itself represents the birth of psychophysics as a scientific discipline of its own. In essence, the WFL traces the perceptive abilities of the human sensory system back to the perception of so-called "just noticeable differences" between two levels of a certain stimulus. For most human senses (vision, hearing, tasting, smelling, touching, and even numerical cognition) such a just noticeable difference can be shown to be a constant fraction of the original stimulus size. For instance with touch experiments have shown that we are able to detect an increase in the weight of an object in our hands if this is increased by around 3%, independently of its absolute value. This is expressed by the differential equation

$$\frac{\partial Perception}{\partial Stimulus} \sim -\frac{1}{Stimulus}. \quad (1)$$

As direct conclusion, the resulting mathematical interrelation is of a logarithmic form and can be used to describe the dependency between stimulus and response/perception over several orders of magnitude [127]. Where this dependency holds in the domain of QoE, typical stimuli have been shown to be waiting and response times as well as audio distortions, i.e. application-level QoS parameters directly perceivable by the end-user. For these reasons, logarithmic relationships have not only been observed in the domains of psychophysics and perceived network performance, but also in the field of economics [101].

Exponential Relationships – The IQX Hypothesis. The second example is the so called *IQX hypothesis* (exponential interdependency of Quality of Experience and Quality of Service) [47,32] which describe QoE as an appropriately parameterized negative exponential function of a single QoS impairment factor. To demonstrate this mapping, iLBC-coded speech samples were sent over a network emulator for adding defined QoS impairments. The resulting degraded samples were recorded and served, together with the original versions, as input to the PESQ algorithm (ITU-T P.862), which automatically calculates the corresponding QoE in terms of MOS values [43]. As a result the authors observed an exponential relationship of the form $MOS = \alpha \cdot e^{-\beta \cdot QoS} + \gamma$ between packet-loss and audio quality scores. The underlying assumption is that within a functional relationship between QoS and QoE, a change of QoE depends on the actual level of QoE [32], implying the differential equation

$$\frac{\partial QoE}{\partial QoS} \sim -(QoE - \gamma). \quad (2)$$

which has an exponential solution.

Both types of relationships confirm the general observation that users are rather sensitive to impairments as long as the current quality level is already quite good, whereas changes in networking conditions have less impact when quality levels already are fairly low. However, they differ in terms of underlying assumptions: the WFL relates the magnitude of QoE change to the current QoS level, whereas the IQX hypothesis assumes that this magnitude of change depends on the actual QoE level. Furthermore, the WFL mostly applies when the QoS parameter equates to a signal- or application-level stimulus directly perceivable by the user (like latency or audio distortion), while the IQX applies in cases of QoS impairments on the network-level which are not directly perceivable (e.g. packet loss). Taken together, both relationships have been found helpful in explaining or obtaining new insights from passive measurements [109] and in the context of studying web applications and waiting times [21,26].

2.4 Applications and Challenges

To summarize the main points of this introduction to QoE, we are now going to reflect on the applicability of QoE by highlighting the different domains or stages of QoE research and related challenges. Taken together, these stages constitute a chain in which the output of one stage provides information and input for the subsequent one as depicted in Fig. 3:

1. **Fundamental Data and Laws of Quality Perception:** By definition, QoE not only demands for conducting extensive subjective test campaigns in order to generate ground truth data (see Section 2.2). As a discipline, QoE also seeks to explain its findings, building on general laws of perception, sociology and user psychology. The challenge here is to perform research in a truly interdisciplinary fashion to conduct valid user experiments that generate accurate and reliable measurements of human quality perception.

2. **Guidelines for System Design and Network Planning:** The insights gained on the fundamental level already allow to develop conclusive guidelines and recommendations for the design and planning of future communication services and networks. Such guidelines typically consist of acceptance thresholds in conjunction with quantified relationships between technical parameters and QoE which often relate to generic relationships such as the WFL or IQX hypothesis (see Section 2.3). The challenge here is to arrive at generalizable, conclusive guidelines that can be successfully applied to the technological settings and application at hand (e.g. dimensioning of an LTE network for mobile broadband services).
3. **QoE Models and Metrics:** Modeling QoE from different perspectives is the result of deep and comprehensive understanding of the fundamental mechanisms underlying quality perception, with the goal to develop quantifiable metrics that describe QoE in a technically accessible way (see Section 2.2). Here the main challenge is to develop robust models that match end-user quality perception as closely as possible for the given service and measurement application (e.g. lab evaluation of a new audio codec).
4. **QoE Monitoring:** beyond suitable metrics, computation of QoE estimates for monitoring purposes requires sophisticated measurement frameworks. These can be realized on top of general purpose network monitoring systems that provide QoS-based input data streams along with contextual information. The main challenges here are computational efficiency and performance as well as data availability, since the monitoring system has to work in real-time and required data sources (e.g. reference signals) might not always be available at the locus of computation (see Section 2.2). In addition, if underlying models rely on application-specific parameters, additional measures like deep packet inspection (DPI) are required to acquire these inputs from network traffic, causing additional performance and privacy issues.
5. **QoE-Centric Network and Service Management:** Finally, beyond monitoring and measurement, above outcomes can be applied to improve the actual management of operational networks (e.g. via QoE-based policy control functions) and related support services, including charging and accounting in real-time. Like with QoE monitoring, a key challenge is to collect and process data in real-time. Furthermore, a central challenge is defining the optimization target, leading to the distinction between user-centric [121] and network-centric [136] approaches⁷. In this context, additional challenges arise when gathering implicit and explicit feedback at the user-level, e.g., how to feed QoE information back to the provider for adapting and controlling QoE [31]. Hence, trust and integrity issues are critical as users may cheat or change behavior to receive better performance.

⁷ Since this chapter focuses on QoE as measurement challenge, a discussion of QoE management approaches is beyond scope. See [115,136] for a comprehensive overview.

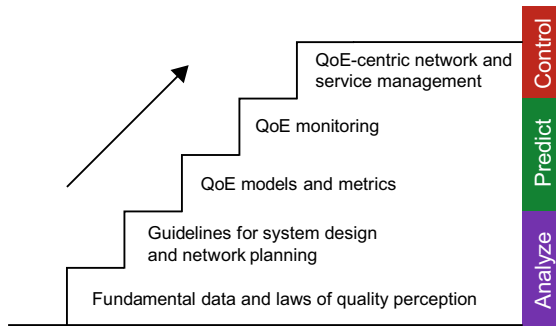


Fig. 3. The different stages of QoE research and application types

2.5 Towards a New Understanding of Quality

This introductory section has outlined the foundations of QoE and its assessment along with its applications and related challenges. It has shown how the concept has evolved out of QoS with the purpose of (re-)introducing user centrality into quality assessment. In this respect, QoE can be considered as new paradigm, since compared to QoS it has a much wider scope and provides a new, fundamentally different perspective on quality in communication ecosystems: beyond purely technical performance dimensions, QoE is about the *human* user's assessment of the different pragmatic and hedonic aspects of a system or service, including also the influence of non-technical factors such as context and the state of the user herself (such as expectations and emotions). The inherent multi-dimensionality and multi-disciplinarity of QoE also requires an enlarged arsenal of assessment methodologies derived from neighbouring disciplines such as psychology and sociology combined with traditional technical measurement methods on network- and application-level in order to obtain a truly new understanding of quality based on reliable and valid results. In this respect, this section has provided an overview and introduction to the fundamentals of QoE. As next step, the following three sections are going to provide an overview of QoE research for voice, audio-visual and web services.

3 QoE for Voice Communication Services

The transition from circuit switched POTS telephone services to packet switched voice services (VoIP) and mobile voice services respectively, has led to an increase in processing elements in the transmission chain and a drastic decrease in bandwidth demand of (narrowband) voice systems. The multitude of processing elements together with the characteristics of the voice carrying IP networks has a time variant and significant effect on user perceived voice quality. Volatile voice

quality is the result. This volatile nature has raised interest in the user perceived quality of voice services, in order to ensure customer satisfaction.

The aim of voice quality evaluation is to identify the influence of the transmission performance of the communication system under test. In order to achieve this goal the following questions have to be answered:

- What are the psycho-physiological principles of voice quality perception?
- How can user perceived voice quality be quantified?
- What objective models do exist for the estimation of voice quality?
- What approaches can be used to monitor voice quality in recent communication networks and what are the related challenges?

In this section we are going to address these questions and discuss how to overcome the associated challenges for voice quality measurement and estimation. We will also describe how the already mature understanding of subjective voice quality can be enriched in order to merge into the holistic QoE concept introduced in the previous section.

3.1 Perceptual Principles of the Human Auditory System

From a purely physical point of view, speech consists of time varying pressure waves emitted through the vocal tract of the speaker having spectral and temporal properties⁸. This signal is transmitted over a medium and received by the ear of the listener where the (physical) signal is converted into a series of perceptual events (cf. [97]). Voice services for mediated communication convert the physical signal emitted by the speaker into an electrical or digital signal which is transmitted over a certain network and reproduced into a physical signal on the listener side again. Typically, voice services add certain impairments to the speech signal, e.g. loss artifacts, coding distortions, delay etc.⁹. Such an impaired signal can then be physically characterized in the acoustic or electrical domain. This technical characterization can be achieved in high fidelity. However, it is much more difficult to assess in which context the two interlocutors currently are and what the state of their conversation currently is. In [42,28] the authors have shown that for a conversational task in two different contexts: strictly limited and unlimited execution time for the task, subjects in the time pressure context were having more speaker changes and were therefore more prone to the negative impact of transmission delays. This means that people being in an urgent need to acquire important information as fast as possible might be more tolerant to the signal fidelity as long as intelligibility and fast interaction (hence low delays) are guaranteed, whereas listeners of an online lecture will be in favour of a high fidelity signal and would not care about higher delays in

⁸ A detailed description of the production and reception ranges of the human vocal system in terms of frequency range, signal levels, maximum resolution of signal dynamics, can be found in [86].

⁹ For a more detailed discussion of impairments of voice communication systems see [78,62,6,97].

the delivery chain. Therefore, focusing on physical signal properties as many of the existing voice quality approaches do (hence covering only the system box in Figure 1 only) assesses solely user perceived quality but falls short in considering interactional, usage situation and service related features which are necessary to migrate from user perceived quality towards the multidimensional perspective of QoE as discussed in Section 2.1.

For identifying a subject's quality perception one can ask the person directly to quantify their experience. This quantification can be achieved along a two dimensional matrix spanned by the *perceptual-affective* dimension and the *subject-objective* oriented dimension (a combination of [97] and [6]) as depicted in Table 3.1. The vertical dimension distinguishes between voice quality tests targeted towards the description of perceived (quality) features (*perceptual*) and a subjective quantification of the overall quality (*affective*). In contrast, the horizontal dimension differentiates amongst the identification of human perception (*subject-oriented*) and sound reproduction capabilities of (voice transmission) systems (*object-oriented*) [97].

Object-oriented voice quality tests are primarily used to identify relationships between system or network parameters and user perceived quality, whereas subject-oriented tests are often used to derive more general models predicting voice quality based on signal characteristics. However, both types of tests share certain characteristics of test setups and evaluation methodologies.

Table 1. The four dimensions of voice quality evaluation (based on [97] and [6])

	Subject-Oriented	Object-Oriented
Affective	Quality perception	Assessment of system quality
Perceptual	Quality features and their perception	Quality features and acoustic or system correlates

3.2 Subjective Assessment Methods and Objective Prediction Models

The aim of speech and voice quality evaluation methods is the quantification of user perceived quality of the communication system under test, in order to use this information for the development of models able to estimate user perceived voice quality based on instrumental measured system and network parameters. Within the following subsections it will be shown what QoE dimensions the discussed approaches are able to evaluate and what is needed to enrich their results towards a holistic QoE assessment.

Subjective Assessment. In general, four methodological categories for subjective voice quality evaluation do exist (cf. [79]):

1. Comprehensibility Tests
2. Multi-dimensional Tests

3. Listen Quality Tests
4. Conversational Quality Tests

As the first two test categories refer to the dimensions of *subject-oriented* and *perceptual* tests and the focus of this book is rather on system evaluation we are going to concentrate on the latter two test categories (listen and conversational quality)¹⁰. The measurement of user perceived quality is usually achieved through the quantification of the user's perception of a voice signal on an opinion scale, with MOS being most common (cf. [77,97]). The most widely used assessment approach are listening (or: listen-only) tests as they allow evaluation of fine grained system differences within short durations. However, this gain in execution speed results in a considerable loss of external validity and QoE dimensions evaluated, resulting in subjective scores which mainly represent the signal (or system) fidelity only. This is based on the fact that listen-only tests by their nature do inherently neglect the context of a conversation and its dynamics which are a prerequisite for the QoE concept outlined in the background section of this chapter (cf. Section 2.1).

In order to overcome these shortcomings conversational tests should be conducted as described in [51,78,58,97]. They feature a high degree of realism by capturing communication dynamics, considering transmission delays and different usage situations through different tasks involved. Nevertheless, the current methodologies for such conversational tests do treat some of these variables as pure experimental variables and do only measure user perceived quality on a certain (MOS) scale instead of additionally measuring conversational surface parameters. Such conversational surface parameters can be used to quantify communication problems induced by transmission delay and can give valuable information about the interactional state of the conversation, usage situations (e.g. degree of interactivity) and user characteristics as shown in [27,28]. Combining such measures with perceptual quality scores would be a big step towards covering all dimensions addressed in Section 2.1, thereby exceeding purely user perceived quality and getting close to QoE assessment. The same line is taken by [75] where the authors discuss different dimensions involved in communication systems QoE from solely perceptual quality up to service quality over longer usage durations.

Objective Models for Voice Quality Prediction. The purpose of objective models is the estimation of the user perceived quality of a voice service. Currently, most of these models do not consider individual user preferences, expectations, the context and conversational dynamics (cf. [77]) but rather estimate voice quality of an average user in a listen only context. An overview of objective models for voice quality, which have been approved by standardization bodies is given in Figure 4. References to these standards will be made in the subsequent paragraphs of this subsection.

¹⁰ Regarding further information on comprehensibility and multi-dimensional tests, we ask the interested reader to consult [62] regarding comprehensibility and [123] for a detailed discussion of multi-dimensional analysis of voice quality.

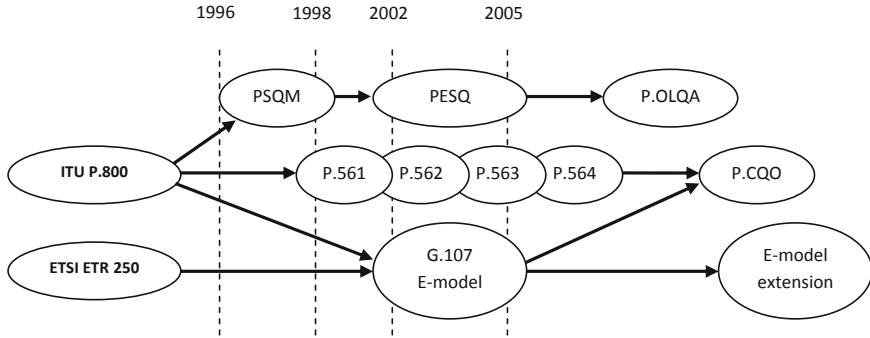


Fig. 4. Overview of standardized objective models within ITU-T and ETSI (from [135])

Signal Based Models: These models analyze speech input signals which have been transmitted or processed by speech processing systems and try to identify and quantify the distortions introduced into the speech signal by means of a psychophysical model. Within signal based models, two approaches can be distinguished: a) single-ended (non-intrusive) and b) reference-based (intrusive) models. Common to both approaches is the fact that due to a lack of delay and echo incorporation, only listen quality estimates can be given.

Single-ended models such as ITU-T P.563 and [73,67] use the distorted signal as only input source which is then pre-processed and passed through a distortion estimation stage and a subsequent perceptual mapping stage (cf. [73]). However, this approach implies that the distortions to be discovered have to be known in advance. Therefore, distortions not known in advance (e.g. new codecs, signal enhancement elements in the system etc.) can not be detected and considered in the output score. Despite this major disadvantage such models are gaining increased interest again as they allow to estimate speech quality in a non-intrusive and passive way within the service network. In contrast, reference based models such as ITU-T P.862 and ITU-T P.863 need the (undistorted) source signal in addition to the distorted signal. Basically, they assess the difference between the source signal and the distorted signal in order to quantify the amount of distortions introduced by the speech processing system (see [78,103,7] for a more detailed description). In contrast to the approach of the single-ended model, this comparison against the unaltered signal allows to identify distortions without prior knowledge about their nature to a certain extent (cf. [77]). As a result, reference-based models achieve higher accuracies in voice quality estimation compared to single-ended models. The most widely used reference based model is the Perceptual Evaluation of Speech Quality (PESQ) model ITU-T P.862. Originally designed for narrowband (NB) speech services but since 2005 also extended to work for wideband (WB) speech services as ITU-T P.862.2. In 2011 its successor the perceptual objective listening quality assessment (POLQA) model ITU-T P.863 has been approved. In contrast to PESQ it has also been trained on

databases including recent codecs and speech enhancement devices and it also operates in a super sideband (SWB) mode (cf. [77]). However, these signal based models¹¹ do only cover the *system* dimension depicted in Figure 1 and do neglect conversational context as well as conversational states of the interlocutors.

Parametric Models: These models originate from network planning as they were intended to predict the voice quality of new networks in advance by incorporating certain (planning) parameter settings. In addition to the listen quality to be expected, such models are also able to incorporate echo and delay related impairments in their quality estimations. Early models of this type are [102,88,54], however the most prominent example of a parametric model is the E-model (ITU-T G.107). It predicts the quality users experience during a voice conversation based on impairment factors such as end-device characteristics, used codecs transport parameters and so forth. The E-model determines a rating R : $R = R_0 + I_s + I_d + I_e + A$ where R_0 is the basic signal-to-noise ratio, I_s takes into account phenomena that occur simultaneously with the speech signal (like the loudness of the speech signal and the side-tone and quantization effects), I_d groups impairments associated with delay (such as, impairments due to echo and loss of interactivity), I_e accumulates the effects associated with special equipment (for example, the use of a low bit rate codec or packet loss), and A is an advantage factor (i.e., a decrease in R-rating a user is willing to tolerate because he or she has a certain advantage, e.g., being mobile). The R-value can then be mapped to MOS scores for speech quality estimation. Revisions of the E-model have improved its prediction accuracy, extended its applicability to WB speech and has lead to work on the implementation of E-model based quality assessment models applicable for online quality prediction [118,119,120]. In respect of the QoE approach described in the background section (cf. Section 2.1), the E-model implements the inclusion of contextual and user related factors (I_d and A) in addition to pure physical fidelity factors (I_s and I_e) as demanded by such a multidimensional QoE conception. However, online assessment models based on the E-model are often not able to include these factors on a per-call basis and therefore forfeit these advantages to a certain extent.

Packet Layer Models: Packet-layer or protocol level models (PLM) are useful when media-related payload information is too (computationally) costly to be analyzed and thus signal-based models cannot be used. The intended application of PLM's is network quality monitoring. Therefore, the focus of such models is on minimal computational effort and input parameters which are easily available in voice communication networks and are typically QoS parameters (e.g. packet loss, packet re-ordering ratio etc.). Examples of such models are described in [20,104,111] and compared in [95]. Although several such models exist, none of them has been standardized yet by an international standardization body. From ITU-T side there only exists a defined methodology ITU-T P.564 for comparing

¹¹ Signal based models can be considered as a special case of listen only models and therefore share the same disadvantages.

the prediction accuracy of different PLM approaches. In addition to these PLM approaches also hybrid approaches do exist which combine estimations of PLM models with estimations from signal based models (e.g. PESQ) such as [118,1]. A more detailed review of such hybrid approaches is given in [77]. Due to the limited information regarding the conversational situation and the involved subjects available on the packet layer, these approaches do mainly assess the signal fidelity of the system and fall short in addressing the other QoE influence factors mentioned in Section 2.1.

From the preceding discussion of objective models it is evident that in light of the holistic QoE concepts all of these models have certain drawbacks. While the E-model as a planning model gets pretty close in considering the majority of relevant QoE dimensions, its online derivatives do implement the incorporation of these dimensions unsatisfactory. The other models discussed do mainly target the pure signal fidelity dimension and therefore are stuck in predicting user perceived voice quality only. However, the case of the E-model shows that objective models which conform to the QoE approach are feasible but more efforts for online extraction of currently not covered features and dimensions is needed in order to feed this information into new models capable of realtime QoE estimation.

3.3 Monitoring and Measurement of Voice Quality and Related Challenges

According to [97], two types of measurement methodologies for monitoring voice quality in communication networks can be distinguished:

Intrusive measurement (offline, active): specific test calls are set up and measurement signals such as noise or speech are transmitted across the network.

From the comparison of the output and input signals, a direct quality estimate can be obtained.

Nonintrusive Measurement (online, passive): at a specific point of the network, a measurement signal is acquired during normal network operation.

From this signal, network or conversation parameters relevant to quality can be derived.

Whereas offline methods are more commonly used in network diagnostics, online methods are widely used for live network monitoring. A major difference between the two categories is the computational complexity (higher for offline methods) involved together with the resulting accuracy (lower for online methods). The accuracy shortcomings of voice quality models used for online measurement are often accepted as the lower computational complexity allows for large scale monitoring. Therefore, we concentrate on these models in the remainder of this subsection. From the three objective model categories mentioned in the previous subsection, parametric and the PLM models are typically used in such a setting. In addition, single-ended signal-based models are sometimes applied for online monitoring, however their computational effort is rather high and poses a critical constraint. For choosing the right model it is important to know where the

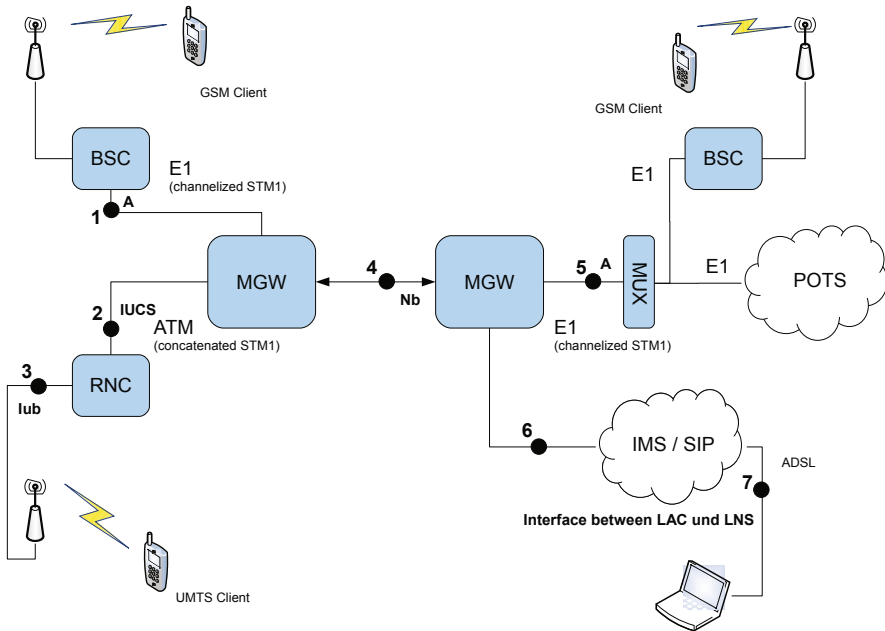


Fig. 5. Potential monitoring points within a typical network topology for voice services

monitoring probe is located, which parameters are available at the probe and what the network topology looks like. Figure 5 shows possible probing points in a simplified network topology with major elements needed for mobile, fixed and VoIP services. Monitoring and associated complexity is of course service dependent, however even the apparently most easiest service to monitor (VoIP) comes with certain pitfalls as the following example should illustrate. In VoIP-only systems one could assume that PLM's or hybrid models are the best choice (e.g. deployed at probing points 1-3, 5-6 and 7 in Figure 5). However, if transcoding is part of the transmission chain (e.g. in a media gateway - MGW), such models fail to properly estimate voice quality since network impairments before the MGW are not detectable in the packet stream after transcoding¹². In such cases a single-ended signal based model (which could be applied in probing point 4 in Figure 5) might be a better approach despite higher computational complexity. These examples demonstrate that despite the maturity of voice quality assessment methods, there is no single 'best' objective model for a given application (such as live network monitoring of VoIP services), nor exists a model that is feasible to estimate QoE rather than user perceived quality only.

The requirements to overcome the addressed limitations towards a new voice QoE measurement paradigm would be: 1) Acquisition of conversational study

¹² This means that when analyzed on the packet-level only, an audio stream arriving at the MGW with packet losses appears to be free from impairments after transcoding.

results, which in addition to common MOS scores also measure conversational surface parameters in order to extract related information on the conversational state, the usage situation and user characteristics, 2) Develop feature extraction methods which deliver aforementioned contextual, user related and service information from the voice signal or the packet stream in an automated fashion, 3) include reliable transmission time (delay) estimators or measurement modules in the measurement frameworks, 4) merge these additionally acquired information into (hybrid) models that are able to estimate voice QoE scores in contrast to solely voice quality scores. Although these requirements are challenging one should not forget that some of them might be relatively easily accessible in communication networks such as e.g. the user's current location as estimation of his current context or his personal usage history as estimator for his conversational personality. Such estimators might serve as shortcuts on the endeavour from voice quality towards voice QoE.

4 QoE for Audio-Visual Services

Audio-visual multimedia constitutes the by far most intensively investigated and technically challenging service category today. The most popular traditional audio-visual service is still broadcast TV, however, on the Internet the amount of traffic generated by on-line video platform like youtube.com, hulu.com or ted.com is growing at unprecedented rates. For example, up to 90% of Google traffic is audio-visual traffic [89].

This growth of audio-visual traffic is driven by the inherent immersiveness and richness of the medium as well as the increasing speed of networks [3]. However, raw video and audio data of a clip is both too large to transmit over an IP-network and also contains lots of redundant information which the human visual system does not perceive. Therefore, audio-visual content is always compressed before transmission, a processing step that can introduce quality impairments. In addition, transmission of the encoded content across the network is not error free and is subject to impairments like losses, packet reordering and delays. Therefore, the final quality at the receiving end is a function of both compression/streaming processes and transmission conditions.

QoE analysis of audio-visual services has to take into account a number of different aspects. The first one is the human visual system. Knowing what people do or do not perceive enables to detect QoE-relevant features and influence factors. In this context, subjective experiments are essential for assessing how relevant a specific feature is. Audio-visual services comprise of both audio and video components. However, since audio-related quality aspects were already extensively discussed in Section 3, this section is going to focus only on visual quality and the human visual system in particular.

4.1 Perceptual Principles of the Human Visual System (HVS)

As already mentioned in Section 2.1, QoE is determined by various factors influencing end-user perception and judgment (see Fig. 1). In case of video quality

analysis, one of the crucial determinants of QoE is the way we see. Without understanding the biological and cognitive processes behind human vision objective metrics or subjective evaluations method cannot be applied correctly. Therefore, the Human Visual System (HVS) is presented here as a background of QoE for video evaluation.

The biological and optical processes behind human vision have been extensively investigated for a long time. This is not surprising, given the importance of this modality for everyday life. The HVS is highly complex, and a clear understanding of how it works in detail has not been achieved yet, particularly as regards the neurological processes behind visual perception. For example, certain effects have been found that create a 3D impression from 2D-only images with the use of perspective lines [98].

When it comes to video compression and QoE analysis, certain physical features of the human eye help a lot to understand the HVS and its limitations. For example, exact reaction, i.e. electric signal produced by cones, of the human eye on different colors (wavelength) is different [30]. Moreover, we see colors at lower resolution than luminance. Therefore, the most common method of encoding video uses twice as much luminance samples/pixels as color samples/pixels [57]. Other significant limitations of the HVS are its lack of ability to detect high frequency details, depth being less important than texture, or unique way to recognize faces [11]. Thus, knowledge of the HVS can help to predict which features of an image or video should or should not be visible. Nevertheless, predicting to what extent influential a particular effect is required enlarging the scope of investigation. A commonly used extension of the HVS is the Visible Difference Predictor (VDP) [24]. The purpose of the VDP is estimating the probability that a particular image feature degradation (for example lack of contrast) is visible. This probability is a function of different image features including the region of the image. Thus, the VDP can be understood as a step from the HVS towards QoE as it provides technical features which can be used to predict QoE.

In order to enable QoE modeling and prediction, it is essential to ask users or observe their behavior under controlled conditions as described in Section 2.2. For visual quality assessment this can be difficult since observing user behavior requires access to a complex system with individual states and influence factors. The common solution to this problem is performing subjective experiment where quality level is controlled precisely as described in the following section.

4.2 Subjective Assessment Methods and Objective Prediction Models

This section provides an overview of subjective assessment methods for video quality along with objective quality prediction models.

Subjective Assessment. Section 2.2 has described subjective experiment methodologies in general. In case of video quality, the recommendations for subjective experiment methodologies have been developed for a long time. The

most relevant recommendations that should be generally adhered to are: ITU-R BT.500 (for broadcast television), ITU-T P.910 (for video quality in case of videotelephony, videoconferencing and video-on-demand), ITU-T P.911 (for audio-visual tests), ITU-T P.912 (for task recognition), and ITU-R BT.1438 (for stereoscopic video). However, when following these recommendations, the challenge remains that the different documents suggest different environmental settings and it is not clear which of them should be actually used in a specific case. Furthermore, environment settings are precisely described - but sometimes in a contradictory way. For example, P.910 describes the room luminance level as ≤ 20 lux and ITU-R BT.500 as ~ 200 lux. Some of the environment settings are also fairly questionable. For example, the room noise level is specified to be the same as for audio tests i.e. following the NR.25 curve. Nevertheless, in case of video quality testing such a room is unnaturally quiet and thus tends to irritate subjects. This shows that the environment as specified by these recommendations can be more disturbing than a typical office or living room.

Finally, also presentation sequencing and rating tasks assigned to the subject vary considerably. In this regard, at least nine different experiment types exist: Double-Stimulus Impairment Scale (DSIS), Double-Stimulus Continuous Quality Scale (DSCQS), Double-Stimulus Comparison Scale (DSCS), Single Stimulus Continuous Quality Evaluation (SSCQE), Simultaneous Double Stimulus for Continuous Evaluation (SDSCE), Absolute Category Rating (ACR), Absolute Category Rating with Hidden Reference (ACR-HR), Degradation Category Rating (DCR), and Pair Comparison (PC)¹³.

Because of this variety of accepted specifications, the comparison or aggregation of results of different subjective experiments tends to be challenging. On the other hand, these different co-existing quality assessment specifications proposing slightly different methodologies also help addressing different research problems and applications correctly. Still, a number of different questions arise in the context of subjective audio-visual quality experiments today, with the following being the most relevant ones:

1. Which rating scale should be used: five point, nine points, eleven points, continuous or other?
2. Which environment settings are the most important to control?
3. Which subjective experiment methodology should be used in case of a specific service or application?
4. How different is 3D quality testing from traditional 2D testing?
5. How different is an audio-visual quality test from a pure video quality test?
6. How does subjective experiment on mobile devices should be specified?
7. What is a valid basis for comparing data between subjective experiments, i.e. how can results obtained from different experiments be compared?

The first question has already been addressed by studies which repeated the same video quality experiment with different subjective scales [49]. The third

¹³ For a detailed explanation of these different sequencing and rating task types please refer to [79].

one was addressed by M. Pinson and S. Wolf in [91]. The fourth question was addressed by Taichi Kawano and Kazuhisa Yamagishi in [64].

The overall conclusion from these studies is that the confidence intervals and mean values of the resulting scores do not deviate significantly different for the different methodologies. Therefore, it is recommended to use the ACR-HR test method and five point quality scale. Nevertheless, many different problems still remain to be solved. Therefore, in practice the design of a specific subjective experiment and reporting of its results has to be accompanied by a precise description of the experiment. Such a description should contain many details, which are not addressed by standards, like the used content type, or are ambiguous like experiment type and subjective scale. An excellent example of such descriptions are the test plans prepared by VQEG, for example [94].

The other questions 2, 5 and 7 are being addressed in [92]. In [92], the results of ten different audio-visual subjective experiments have been compared. The obtained results are not solid enough to make strong conclusions, but based on the subjective experiments' comparison it seems that environmental settings are not so important than one could expect. Audio-visual tests lead to stronger diversity (i.e. variance) of user ratings than video-only quality test, because subjects seem to perform the mental fusion of audio and video quality in different ways.

The briefest answer to question 6 is that for quality evaluation on mobile devices, simple rewriting or reuse of existing test standards and protocols is not possible, therefore the investigation of new approaches is required (cf. [16]). Moreover, mobile devices and their capabilities are changing rapidly, e.g. smartphones are much more sophisticated and bigger in terms of screen resolution than just two years ago. Additionally new devices, e.g. tablets, become popular. Again, any methodology setup for mobile phones or TV screens cannot be simply reused in case of tablets.

Another important and relatively new topic is stereoscopic 3D streaming (question 4). In case of 3D video, certain aspects known from 2D video quality have to be found to apply (e.g. resolution, frame rate, color perception impact). On the other hand, numerous additional aspects like viewer fatigue, naturalness and depth perception have to be addressed differently. However, the only standard on this subject ITU BT.1438 does not address those questions with sufficient level of detail [19].

The above challenges show how important is to plan and describe a subjective experiment precisely in order to ensure reliability of measurements and comparability of the results. Furthermore, in order to draw the right conclusions on the results it is also essential to correctly describe all factors influencing results such as subjects, sequences, instruction given to subjects and room conditions.

When it comes to data analysis, a critical aspect is that participants have provided answers on a continuous or interval quality scale. The point is that the opinion scales typically used are an ordinal scale i.e. they have an order but not a defined distance between scale points (e.g. the semantic distance between 'good' and 'fair' does not necessarily equal the distance between 'fair' and 'poor'). The

fully correct way of analyzing such opinion data and avoiding the problem of varying semantic distances is to extract each answer's *probability*, not simply the mean value (as used for MOS), an issue extensively explained in [61]. Analysis of probabilities instead of the average only is also closer to the very idea QoE which is about into account individuals' opinions: a result of the type "60% of users rate the service as good or excellent" reveals how many individual users like the service. On the other hand, overly focusing on mean values leads to conclusions like "the average user judges the service 3.0 on a MOS scale". However, a MOS of 3.0 can be obtained from answers widely scattering i.e. obtaining five (excellent) or one (bad) or from users answering only three (fair). Indeed, such high rating diversity (or high variance) situations are quite likely to occur in audiovisual tests. For example, with poor video and excellent audio, some users still rate the service as good or excellent since they focus on audio while others focusing mostly on video will describe it as poor.

All presented problems address only quality evaluation problem. More studies are needed to answer real QoE problems. For example in [14] everyday live situation was introduced to the quality test showing clearly that social and emotional aspects have strong influence on the QoE evaluation. Introducing such additional effects is difficult from the subjective experiment point of view. It is nearly impossible to be measured by the quality aware network. It limits practical usage of real QoE results which takes into consideration much more than just compression and network distortions. On the other hand, for some applications it is possible to use such more detail context. In [134] recompression algorithm is driven by surrounding contextual influences.

Objective Models for Video Quality Prediction. In general, the goal of QoE modeling is to find an objective metric which for the given scope of quality features and influence factors strongly correlates with data from user studies. This function should be based on measurable features of the system (see Section 2.2 and Fig. 2). The most common and (most simple) classification of video metrics is full reference (FR), reduced reference (RR) and no reference (NR), where reference means ability of accessing the original sequence (see also Section 2.2). Another classification discriminates according to the level of access of information. This can be network headers only, bitstream i.e. partly decoded sequence (for example moving vectors), picture (=signal) analysis or any information. The latter metric using all available information is called hybrid [114]. A more detailed presentation and description of different types of video metrics can be found in [132], on which Fig. 6 is actually based.

As of today, the most advanced metrics are based on full reference pixel domain information. There numerous different metrics exist [124] but the most popular ones are PSNR, SSIM and VQM. In addition, J.341 the latest standardized metric proposed by SwissQual should be added to this list. PSNR is the simplest possible metric taking into account only signal to noise ratio. However, it is also very easy to produce results that strongly deviate from human perception since adding just a single pixel shift influences strongly the obtained score. On the other hand, by adding simple mechanisms like searching for the best

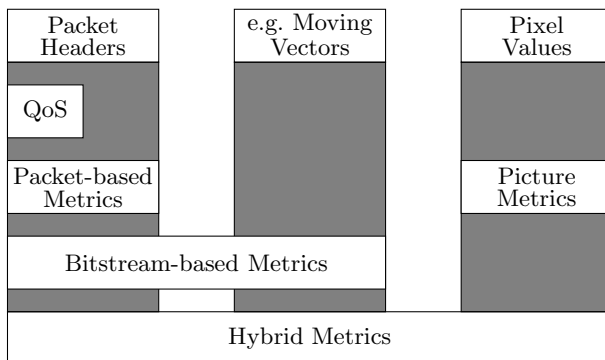


Fig. 6. Different categories of objective video quality metrics (based on [132]). The range of the network QoS domain has been added for illustration purposes.

fitting and luminance normalization, PSNR is changed to a metric performing so well that it is difficult to outperform it [93]. SSIM is much more complicated than PSNR [125]. It is based on the HVS and was created for image compression analysis just like PSNR. The extension of SSIM for video is also published [126]. The last metric is VQM which looks like a simple linear combination of several image features [133]. Nevertheless, it was created based on many different subjective experiments and it is shown that VQM performance is very good and it became a part of a ITU-R standard. The last standardized full reference metric was proposed by SwissQual which showed to be the best at the HDTV test provided by VQEG and standardized in ITU-T Recommendation J.341.

As regards input signals required, full reference (FR) metrics need both the original sequence and the decoded distorted signal. Such limitation makes it very difficult to use it for monitoring purposes in a real-world telecommunications or IPTV network. Even assuming that computation resources are not an issue and a single channel is used to broadcast a test sequence, monitoring this specific channel does not guarantee that the obtained quality is the same for other channels. Therefore, using FR metrics is difficult in practice.

In order to solve this problem, reduced reference (RR) metrics are proposed. In case of pixel domain analysis again HDTV test was able to identify a metric, proposed by Yonsei University, which should become a standard soon. Nevertheless, the problem of RR metrics is that an operator or service provider has to compute the coefficients to be transmitted as the reduced source information.

Finally, no reference (NR) metrics are particularly relevant to practical monitoring applications since the quality is obtained based only on the impaired stream under test. NR metrics can just take into account a few network QoS parameters or be very complex by additionally processing packet headers and pixel domain information. Metrics taking into account both pixel domain and different layers headers or some decoded information, like moving vectors, are called Hybrid metrics. These kinds of metrics are now evaluated under the VQEG Hybrid test plan and if enough good results are obtained, a new standard will

be released. Moreover, the VQEG JEG-Hybrid group evaluates different hybrid metrics and in order to create a joint effort hybrid metric [114]. This is an especially promising initiative since currently, hybrid or bitstream metrics are developed by many different research institutions and companies. In case of 2D images some interesting results have already been published [33,48,34].

4.3 Monitoring and Measurement of Video Quality and Related Challenges

In practice, video quality is monitored mostly for IPTV service quality assurance purposes. Therefore, today's most relevant metrics from a business point of view are non-reference metrics. Video QoE monitoring is supposed to help network operators to understand where, from both network and quality point of view, is the problem. However, currently available video quality metrics actually support problem recognitions fairly well. However, they much less support problem understanding and root cause analysis. Therefore, better diagnosis of error and origins of quality defects is one of the most important QoE challenges in this domain.

Recently ITU-T SG 12 released two standards which are targeting IPTV monitoring use cases called P.1201 and P.1202. They are both bitstream models which are especially useful for monitoring since packet-level analysis is much easier to scale than decoded video stream monitoring. P.1201 is a no-reference audiovisual packet-based metric (see Fig. 6) limited to single H.264 decoder. P.1202 is similar but it can also decode the signal which classifies it as Hybrid Metric (see Fig. 6).

Challenges faced by the video quality community typically go beyond the understanding of reasons behind certain quality degradations. The questions cover different aspects of general QoE challenges described in Section 2.4 converted to a list of challenges for audio-visual QoE includes (but is not limited to):

- Developing a standardized non-reference hybrid metric which is not limited to a single decoder,
- Linking QoE metrics results to user behavior (i.e. viewing times or cancellation rates as function of QoE), and
- Linking metrics with particular reasons behind quality impairments.

The range of video services and use cases is still growing and thus the list of new related QpE challenges can be supposed to grow equally. On the other hand, to some problems in this domain fairly advanced solutions do exist. For example, today's standards describing FR metrics are fairly mature, making it very difficult to propose a new FR metric since it has to be significantly better than the standardized ones. Another solved problem is the lack of high quality 2D test sequences. Thanks to www.CDVL.org, a large set of uncompressed 2D sequences is finally freely available.

Finally, when comparing the fields of video and audio quality (see Section 3) it becomes clear that the models obtained for audio quality metrics are much

more advanced and comprehensive (cf. [123,76]). Thus, future research should aim to reduce this gap by joint development of metrics for both modalities as well as their integration in multimodal scenarios (cf. [79]). Furthermore, it is important that QoE is not only relevant for voice and audio-visual multimedia services, but also for interactive services on the web.

5 QoE for Web-Based Services

In the past decade the Internet changed completely the communications world as well as daily life of most people. The Internet offers an increasing diversity of distributed applications and services that are delivered to the end-user over the web. The major web-based services comprise information dissemination, video delivery like YouTube video streaming, as well as collaboration and interaction between humans like social networking (e.g. Facebook). These applications have in common that their implementation utilizes the HTTP protocol on application layer, as well as the TCP/IP protocol on transport and network layers, respectively. The HTTP protocol is based on a request-response pattern between server and client. Hereby, the term 'client' denotes an instantiation of the application client software at the end user device. For the end user, this request-response pattern results into a certain amount of *waiting time before service consumption*. For example, when the end user requests a web page in her Internet browser, it takes some time until the web page is downloaded, rendered and displayed at the end user device. Below HTTP, the TCP/IP protocol offers reliable transport of data between the client and server in general. In case of insufficient (server or network) resources, the end user consumes more data than the server (or the network) is able to deliver. As a result, those network impairments are perceived as *waiting times during service consumption* by the end user due to the reliability of the used transport protocol. For example, HTTP-based video streaming suffers from insufficient network data rates when the video contents are not delivered fast enough to the client, such that the video stops for a certain amount of time until the video buffer is filled again and the video playback is restarted. Hence, the usage of TCP/IP makes the end user perceive network impairments as waiting times¹⁴. The impact of waiting times on QoE for HTTP-based video streaming will be discussed in the next chapter "Internet Video Delivery: From Traffic Measurements to Quality of Experience" in more detail.

As concerns this chapter, the focus lies on the perception of waiting times before service consumption by the user. To this end, it first briefly revisits human time perception from a psychophysical perspective (Section 5.1). Then, simple waiting tasks in web-based services are considered to demonstrate how to assess and quantify QoE as basis for related QoE models (Section 5.2). The investigation of more complex services such as web browsing poses however several challenges on user level, i.e. how waiting times are perceived by the end user, as well as on technical level, i.e. how to monitor waiting times in practice. Together

¹⁴ In contrast, unreliable transport protocols make the end user perceive network impairments as degraded video quality which is discussed in the previous Section 4.

with the given diversity of web-based applications ranging from video streaming to web browsing, these challenges characterize the complexity of quantifying and monitoring QoE for web-based services (Section 5.3).

5.1 Perception of Waiting Times by Web Users

The application of stimulus-response models for QoE measurement scenarios has turned out to allow important insights into the relationship between objective measures like technical QoS and subjective QoE metrics. In the context of waiting time perception, the objective measure is given by the waiting time itself. However, the resulting QoE of this waiting time includes different facets like time estimation by the subject, perception of durations, or the underlying timing systems in the human brain. By its nature, time cannot be a direct stimulus but is a certain duration between electrical stimuli signals of the nervous system. This requires the transformation from physical signals into electrical signals in the nervous system via a sensory organ. Due to the different (temporal) properties of different sensory organs the temporal resolution differs for stimuli of different modalities. E.g., auditory stimuli are more precisely processed on a temporal level compared to visual or tactile stimuli [38]. For web-based services, mainly visual stimuli exist on a temporal level, e.g. when rendering a web page.

For analyzing user satisfaction as a consequence of perceived duration, [108] states that this is only meaningful when the perceived duration is compared to a tolerance threshold serving as a reference. If the perceived duration is shorter than the tolerance threshold, the user interprets that as fast and decent. Conversely, if the duration is perceived as longer than the tolerance threshold, the user interprets the duration as slow and insufficient. The value of this tolerance threshold is influenced by the usage context, personal factors, past experiences etc. [108]. As a consequence, different web-based services will lead to different QoE models depending on waiting times as shown in the next section.

In this context, the question arises which underlying fundamental relationships between waiting times and QoE may be applicable (see also Section 2.3). Initial work on psychophysical principles in human time perception has been conducted by [29] in 1975, where a relationship between the magnitude of the error of time estimations and the duration of the sample length to be estimated has been identified and attributed to Steven's Power Law [116]. Successive work by [2] extended these results and added other models including the Weber-Fechner-law while [66,37] set out to identify the minimal achievable error for time estimation based on the aforementioned models. They came to the conclusion that the relationship between estimation error and stimuli length is constant, which is essentially a version of Weber's law where the estimation error (termed 'Weber Fraction') is equivalent to the 'just noticeable differences' between two levels of a certain stimulus. For most human senses (vision, hearing, tasting, smelling, touching, and even numerical cognition, such a just noticeable difference can be shown to be a constant fraction of the original stimulus size. As a straightforward conclusion, the resulting mathematical interrelation is of a logarithmic form and can be used to describe the dependency between

stimulus and response/perception over several orders of magnitude [127]. Further, [5] shows that for the subjective evaluation of waiting times on a linear scale a logarithmic relationship does apply.

5.2 Assessment and Models for Web QoE

In general, the term Web QoE stands for the Quality of Experience of interactive services that are based on the HTTP protocol and are accessed via a browser [44]. The most prominent application examples of this category are surfing the web, downloading files (e.g. mp3 songs) and handling e-mails. In the context of interactive data services a number of studies and guidelines exist: [53] defines maximum waiting times for interactive data services unfortunately without empirical evidence how violations of these guidelines do impact user perception. [83] gives similar recommendations about which waiting times are acceptable to the perceived interactivity of interactive data services. However these recommendations do not differentiate between different kinds of such services.

As concerns web browsing, it has been widely recognized that in contrast to the domains of audio and video quality, where psycho-acoustic and psycho-visual phenomena are dominant, end-user waiting time is the key determinant of QoE [83,9,82]: the longer users have to wait for the web page to arrive (or transactions to complete), the more dissatisfied they tend to become with the service. In the following, we focus on web browsing and how to assess its QoE.

From Pages to Sessions. From a technical perspective, a web page is an HTML (Hyper Text Markup Language) text document with references to other objects embedded in it such as images, scripts, etc. While HTTP (Hyper Text Transfer Protocol) constitutes the messaging protocol of the Web, the HTML describes the content and allows content providers to connect other web pages through hyperlinks. Typically, users access other pages or new data by clicking on links, submitting forms. Within this basic paradigm, each clicked link (or submitted form) results in loading a new web page in response to the respective HTTP request issued by the user, resulting in a new *page view* whose QoE is characterized by the time the new content takes to load and render in the browser. Furthermore, the surfing user typically clicks through several pages belonging to a certain web site and of course also occasionally changes sites as well. In this respect, user's web *session* can be characterized by a series of page view events and the related timings of the stream of interactions. During the experience of a web session, psychological factors like memory effects may appear as key influence factors on web QoE and have to be taken into account in the modeling process, e.g. by using hidden memory Markov models [44].

From Request-Response to Flow Experience. The speed and fluidity of the browsing experience has been shown to depend on a number of factors, particularly on QoS parameters of the underlying network. In particular, large packet delay or low bandwidth are well known to cause long loading times of

objects and thus unacceptable completion times of page views (cf. [9,13,4]). In this respect, the time elapsed between the URL-request (e.g. caused by a click on a link) and the finished rendering of the Web page, referred to as page load time (PLT), is a key performance metric (see Figure 7b). Another relevant metric is the duration from request submission until the rendering of the new page starts, i.e. when the user receives the first visual sign of progress [87,23]. In dedicated lab studies, these page view centric metrics have been shown to directly correlate with QoE [55,50]. Thus it seems that waiting times related to the progress of page views are sufficient for predicting Web QoE.

However, several web studies confirm that web browsing is a rapidly interactive activity (cf. [128,112]). Even new pages with plentiful information and many links tend to be regularly viewed only for a brief period - another reason to offer concise pages that load fast [83]. Thus, users do not perceive web browsing as sequence of single isolated page retrieval events but rather as an immersive *flow* experience (cf. [112]). The notion of flow implies that the quality of the web browsing experience is determined by the timings of multiple page-view events that occur over a certain time frame during which the user interacts with a website and forms a quality judgment.

In addition, since during rendering page elements are typically displayed progressively before the page has been fully loaded, the user's information processing activity tends to overlap with the page load phase. The screen real-estate of the browser windows tends to be limited, with pages appearing to be complete before even having been fully loaded. As a consequence, the user's perception of waiting time and latencies becomes blurred by the rendering process itself (which in turn is strongly influenced by page design and programming) [112,68].

Subjective Testing Methodologies for Web Browsing QoE. In contrast to audio and video quality assessment methodologies, where several accepted and even standardized testing methodologies exist, there is far less guidance for proper testing methodologies for web browsing QoE. One main difference towards audio and video assessment methods is the difference in task and related user behavior since the surfing user typically does not issue a single request which is then answered by a short single media experience, but rather a series of such request and responses is typical for web page usage. Figure 7a depicts the two request - response patterns involved in web browsing where $T_1 + T_2$ or $T_3 + T_4$ respectively, characterize the waiting time for one page view. A web session consists of several of such waiting times which are typically of different length (cf. Figure 7b).

A testing methodology for web browsing QoE must therefore ensure that such request - response patterns are issued throughout an evaluation. In order to achieve this goal two different approaches can be distinguished: 1) a defined number of requests or 2) a defined duration one web session. Approach 1) as used in [55,32,44] demands two requests and following responses and page views as depicted in Figure 7b (therefore addressing just a subset of page views of a whole web session as indicated by the zoom beam in Figure 7). After completing

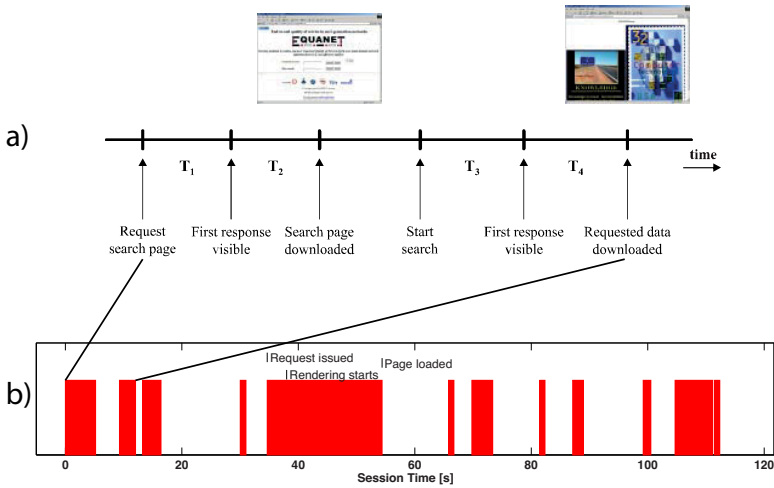


Fig. 7. a) Waiting times related to request-response patterns in web browsing [55] and b) Web session as series of page views with different waiting times

the user is then prompted for his quality rating on an ACR scale. The independent variables here are the $T_1 + T_2$ and $T_3 + T_4$ times. Contrary, approach 2) which was utilized in [106,105,99] uses pre defined session times. For each session the user is asked to execute a certain task on the given webpage while network parameters (e.g. downlink bandwidth, round trip time) are varied as independent variable. After the session time is elapsed the quality rating is gathered. Whereas approach 1) considers the overall session time as independent variable against which the MOS are plotted, approach 2) uses network level parameters as independent variable which then influences the waiting times for each request - response pair. While the latter approach guarantees a more realistic web browsing experience for the user resulting in a series of waiting times (cf. Figure 7b) the user is exposed to, the former approach allows exactly controlling waiting times. Depending on the aim of the web browsing study to be conducted one has to decide for one has to weigh advantages and disadvantages of these two approaches.

Existing Web QoE Models. Models for Web QoE have to take into account temporal stimuli to quantify waiting time perception by web users. One approach is to directly relate Web QoE to those waiting times being a key performance metric. Reflecting the three different instrumental quality models described in the background in Section 2, a Web QoE model based on waiting times can be considered as a degenerated signal-based model with waiting time as 'perceived signal'. Another approach considers network parameters like delay and bandwidth to quantify Web QoE without explicitly considering waiting times. From that perspective, Web QoE is described as parametric model (e.g. on behalf of planning parameters) or packet-level model (for monitoring purposes on behalf of network characteristics measurable on packet-level).

Due to the diversity of existing web services one has to differentiate between web browsing and file download.¹⁵ For *single web pages*, a first signal-based model has been presented in [99] which provides a logarithmic relationship between waiting times in terms of web page load times and web QoE. Experiments on file downloads are further considered, where users are essentially asked to perform waiting tasks (clicking on a link in order to start the download of a mp3 or zip file, and attending the completion of the download) for different downlink bandwidths. From this test setup, one could easily argue that, based on the indirect dependency of waiting time on offered bandwidth, in fact the user's perception on plain waiting time has been measured which turns out to be of logarithmic form. Further evidence for the applicability of the Weber-Fechner law for Web QoE, leading to a logarithmic relationship between Web QoE and waiting times is given in [26]. The subjects were asked to browse through a picture album or to perform google searches. In both cases the request for the next picture and the search result were delayed for a certain time, respectively. The results taken from [26] and shown in Figure 8(a) indicate that the assumed logarithmic relationship holds true for file download, web browsing (i.e. performing google searches), and photo album browsing. For mobile services, similar relationships for Web QoE and waiting times were derived in [84] for web browsing, e-mail, and file downloads.

The parameters of the curves describing the logarithmic relationships differ from task to task, which can be explained by the existence of different tolerance thresholds for different tasks. It has to be noted that duration estimation of waiting times below 0.5s have to be treated different from longer waiting times [39]: QoE reaches saturation for small waiting times and thus logarithmic relationships only applies above the saturation point, i.e. for noticeable waiting times.

However, monitoring of page load times for web browsing is a difficult task, as it will turn out in Section 5.3. Therefore, parametric or packet-level models of Web QoE are highly relevant, too. In [26], participants were also asked to browse different web pages while the downlink bandwidth was manipulated. Figure 8(b) shows the gathered respective ratings for each bandwidth setting and the corresponding logarithmic fitting in dependence of the downlink bandwidth. However, it can be seen that the logarithmic fitting does not match the MOS values very well. In contrast, the IQX hypothesis as introduced in the background Section 2 fits quite well yielding to an exponential relationship between MOS and downlink bandwidth. The consequences of this result will be discussed in the next section on challenges in monitoring and measuring web QoE.

While the above models for web browsing consider a single page, a web user typically browses through several pages within a web session. ITU-T Rec. G.1030 models QoE of a Web search task as a function of (weighted) session times. The

¹⁵ Other web services like video streaming over the web are out of the scope of this chapter. QoE models for Internet video streaming are discussed in another chapter of this book "Internet Video Delivery: From Traffic Measurements to QoE for YouTube".

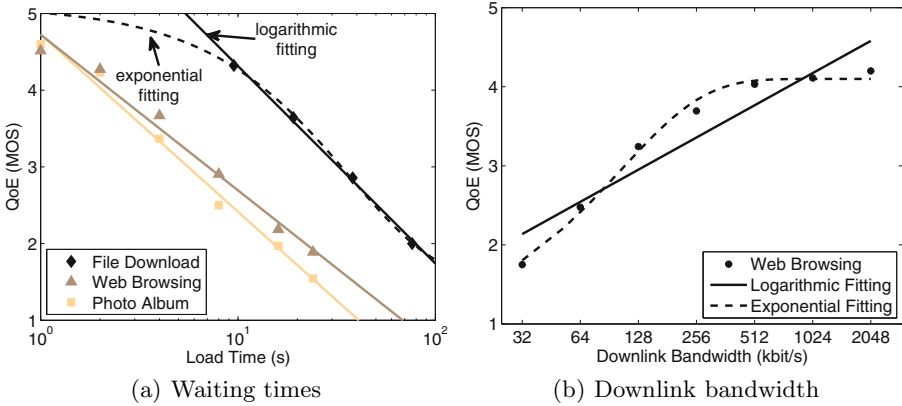


Fig. 8. Relationships between QoS parameter and QoE for web services

corresponding fitting function between the weighted session time and Web QoE is given in [32]. With the surfing user clicking on several web pages in a row, the complexity of Web QoE further increases. [44] has introduced the memory effect to the field of Web QoE modeling, being motivated by the fact that a person’s current experience of service quality is shaped by past experiences. The test scenario used considers a user sequentially browsing the pages of an online photo album. The results show that, although the current QoS level clearly determines resulting end-user quality ratings, there is also a visible influence of the quality levels experienced in previous test conditions. In particular, in addition to the current QoS level the user experienced quality of the last downloaded web page has to be taken into account. The implications of web QoE assessment and modeling are twofold: firstly, the design of dedicated Web QoE studies are required to quantify the impact of the memory effect on QoE, i.e. to consider web sessions instead of single web pages. Secondly, time-dynamics and the internal state of the user (that both manifest in the memory effect) are essential components of the web experience and thus need to be adequately reflected in Web QoE models.

Relation between Acceptance and Web QoE. Besides quantifying QoE, an ISP or service provider may also be interested in the actual acceptability of a web service. In this context, the term *acceptability* refers to a “binary measure to locate the threshold of minimum acceptable quality that fulfills user quality expectations and needs for a certain application or system” as discussed in [107]. Therefore, the relation between acceptance and Web QoE is considered shortly in the following. In the subjective studies in [26], the users were asked a) to rate QoE, i.e. their degree of satisfaction on a 5-point ACR scale ranging from ‘bad’ to ‘excellent’, and b) to indicate their acceptability using a binary yes/no measure for answering the question: “Would you continue to use the service

under these conditions or not?”.¹⁶ The corresponding QoE results were plotted in Figure 8(a) for different web services that are file download, web browsing, and browsing through a photo album.

Figure 9 relates the obtained QoE ratings to the percentage of users who accept the service. It can be seen that 90 % of the users accept already a fair quality with a MOS value about 3 for the considered web services. Thus, monitoring the users’ acceptability of a web service means to check if the found threshold is exceeded when applying a certain QoE model for monitoring.

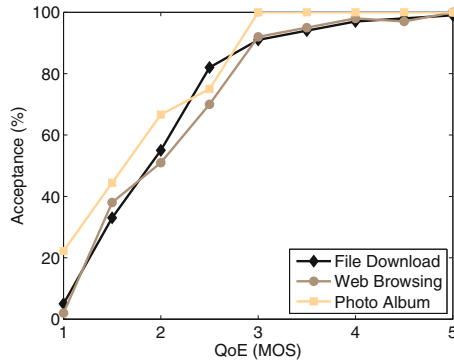


Fig. 9. Relation between acceptance and QoE for web services

5.3 Challenges in Monitoring Waiting Time Perception

The complexity of Web QoE models makes an accurate monitoring of waiting time perception by means of objectively measurable parameters fairly challenging. Furthermore, web QoE comprises a large variety of different applications which require different QoE models for each service like file download, mail, HTTP-based video streaming or web browsing. Especially, the diversity of web sites and their implementations additionally increase the complexity of the problem. Although the same fundamental relationship may be applied like the Weber-Fechner law, web QoE models for particular web-based services and particular web sites have to be derived accordingly. Furthermore, the consideration of flow experience and web sessions may require extra efforts to incorporate resulting influence factors like the memory effect.

In this section, we discuss challenges and practical issues only when measuring the waiting times for web pages. With the Weber-Fechner law, the waiting time may be translated into subjective user perception directly, i.e., waiting time as stimulus is related to user perception. Since waiting times cannot be measured directly, the first idea is to compute the waiting time in terms of page load time

¹⁶ More details on acceptability and QoE as well as on the technical implementation of the test setup can be found in [107].

by means of the download bandwidth and page size. However, this leads to the following issues. In contrast, bandwidth is not a stimulus in a strict psychological sense. Hence, the Weber-Fechner law can only be applied if there is a linear relationship between bandwidth and time.

[26] shows that the logarithmic fitting does not perform well for web browsing. Consequently, the relationship between waiting time and bandwidth is not linear. However, even the relationship between objectively measurable page load time and bandwidth is not linear due to the complexity and interactions of the HTTP and TCP protocol with the network performance (e.g. impact of high bandwidth-delay product on TCP performance; impact of TCP's slow start, congestion and flow control on loading times of small pages; HTTP pipelining, cf. [8]). In particular, a web page consists of several objects leading to chatty client server communications over HTTP. To this end, HTTP pipelining allows simultaneously downloading HTTP objects of the same page. This leads to complex, non-linear models of *network-level page load times* for entire web pages. Furthermore, in addition to the network page load time, the local machine rendering and displaying the web page requires a certain amount of time. Hence, the *application-level page load time* differs from the network PLT and may vary dramatically for different types of web pages, e.g. due to the actual implementation, the used plugins, etc.

As we have already seen, there are several factors yielding to non-linear relationships between bandwidth and (network and application) page load time. However, the Weber-Fechner law considers waiting times, i.e. user perceived PLTs. In psychology, it is well known that subjectively experienced time and objective physical time differ [39]. In addition, in web browsing a page might appear to the end-user to be already loaded although page content is still being retrieved, due to the progressive rendering of the browser, asynchronous content loading (AJAX) and the fact that pages are often larger than the browser window itself. To assess the resulting differences between perceived subjective PLT and application-level PLT, we additionally asked participants in dedicated

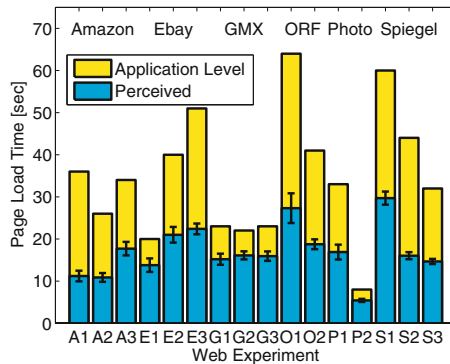


Fig. 10. Perceived subjective vs. application-level PLT for different pages

tasks to mark the point in time when they considered a page to be loaded, i.e. the subjective PLT in [26]. Figure 10 shows the application-level PLT and the subjective PLT for different page types (and three different pages within each type, e.g. front page, search results and article detail page for Amazon). It can be seen that there are large differences between technical and perceived completion time, with ratios ranging from 1.5 up to 3 (where 1 would be the exact match between subjective and application level PLT).

Summarizing, all these different aspects lead to practical issues and challenges to measure or estimate the waiting time as input for the web QoE model.

6 Conclusions and Key Challenges

In this chapter we have provided an introduction to the field of Quality of Experience (QoE) as well as an overview of relevant QoE research themes for selected service categories. The discussion of its evolution along with related research issues has revealed the most defining characteristic and key challenge that differentiates QoE from related concepts like QoS: its strictly user-centric perspective which also causes its multi-disciplinarity, multi-dimensionality (based on the influence factors and quality features mentioned in Section 2.1) as well as its wide applicability beyond pure telecommunications. This level of breadth is accompanied by an equally remarkable level of depth, a consequence of the highly complex nature not only of today's networked multimedia systems but also of the cognitive quality perception processes happening within human users as well. In this light, while the research community seems to be on the path towards an agreed definition and conceptualisation of QoE as "*degree of delight or annoyance of the user of an application or service*" (cf. [17]). However, being able to actually 'measure' QoE with the same ease and accuracy we are measuring QoS today might – despite all ongoing efforts – remain an elusive, maybe only asymptotically reachable goal in the foreseeable future.

In this respect, quality assessment for *voice services* can be considered as the most advanced field compared to the other service categories discussed in this chapter. This is mainly due to the long tradition of voice quality research and the manageable scope and complexity of auditory perception (in comparison to e.g. vision or human-machine interaction). However, we have shown that even in the fairly mature field of voice quality, established evaluation methodologies and models fall short on relevant dimensions of the QoE concept (as influenced by system, user, context) and rather focus on signal fidelity as delivered by the technical system. The next step towards QoE by overcoming these deficiencies would be the integration of additional aspects pertaining to conversational dynamics and contexts. If related parameters can be derived and incorporated in models, such enhanced voice quality metrics would feature higher prediction accuracy and conversational realism.

As concerns *video services*, a considerable number of subjective QoE test methodologies has been standardized so far. However, they mostly focus on the assessment of picture quality in very specific contexts such as desktop PC or

living-room IPTV viewing - with non-trivial adaptations being necessary for new contexts and devices. Similarly, the development of video metrics has shown that models require considerable improvement before they can be used as effectively as their voice quality pendants: various metrics based on pixel- or bitstream-level input information have been standardized recently, nevertheless the universal applicability of these metrics is still limited due to the inherent complexity of human vision and the cognitive processes behind. Thus, the greatest challenges in the short term relate to the advancement of objective metrics towards maintaining high accuracy of prediction in terms of picture quality independent of e.g. type of visual content shown. Mid- to long-term challenges relate to multi-modal fusion as well as using objective metrics for QoE-based optimization of network resource allocation. However, the accelerating rate of service innovation increases the challenge since current metrics do not simply migrate from one service to another without losing validity.

As the currently least mature domain, *web QoE* remains a dynamically evolving field. Although generic relationships between waiting times and web QoE corroborated by results from psychological research on human time perception have been discovered, the inherent cognitive and technical complexities related to this service category have not been sufficiently addressed yet. In particular, web QoE models need to adequately capture the highly interactive and immersive nature of web surfing. To this end, appropriate standards for subjective assessment methods still need to be developed.

Existing web browsing QoE models however tend to consider either very simple usage scenarios or reduce surfing to a simple request-response transaction with a given waiting time. Furthermore, the waiting time perceived by human users does not show a clear correlation with the technical, objectively measured page load time. So far, web QoE models do not support the mapping of typical network QoS parameters such as available bandwidth, packet loss or delay to QoE. As a consequence, the realization of web QoE monitoring currently faces several practical issues and challenges already at the input-level, e.g. how to measure or estimate the waiting time as input for web QoE metrics.

In general, we have shown for all the service categories discussed that the introduction of Quality of Experience becomes progressively more challenging the higher one climbs the staircase from subjective assessment to objective modeling up to QoE monitoring (as depicted in Fig. 3). In this context, the largest measurement challenges arise when it comes to the actual fusion of the subjective assessment of people's perception with the objective measurement of packets and signals. Nonetheless, we are convinced that despite these challenges QoE has the potential to become the future guiding paradigm for understanding, measuring and managing quality of applications and services in future communication ecosystems.

Acknowledgements. This work has been supported by the COST Actions TMA IC0703 and QUALINET IC1003, as well as and the project G-Lab, funded by the German Ministry of Educations and Research (Förderkennzeichen 01

BK 0800, G-Lab). Furthermore, this work has been supported by the projects ACE 2.0 and U-0 at the Telecommunications Research Center Vienna (FTW) that is funded by the Austrian Government and the City of Vienna within the competence center program COMET. In addition, this work was also funded by Deutsche Forschungsgemeinschaft (DFG) under grants HO 4770/1-1 and TR257/31-1 (project “OekoNet”) as well as by the Polish national project 647/N-COST/2010/0. The authors alone are responsible for the content of this chapter.

References

1. Al-Akhras, M., Zedan, H., John, R., Al-Momani, I.: Non-intrusive speech quality prediction in voip networks using a neural network approach. *Neurocomputing* 72(10-12), 2595–2608 (2009), <http://www.sciencedirect.com/science/article/B6V10-4V1TXP7-1/2/c089682667034eb3874ab651e69e21c1>; Lattice Computing and Natural Computing (JCIS 2007) / Neural Networks in Intelligent Systems Designn (ISDA 2007)
2. Allan, L.G.: The perception of time. *Attention, Perception, & Psychophysics* 26(5), 340–354 (1979)
3. Anderson, C.: TED Talks: How web video powers global innovation, http://www.ted.com/talks/chris_anderson_how_web_video_powers_global_innovation.html
4. Andrews, M., Cao, J., McGowan, J.: Measuring human satisfaction in data networks. In: *Proceedings of INFOCOM 2006*. IEEE (2006)
5. Antonides, G., Verhoef, P.C., van Aalst, M.: Consumer perception and evaluation of waiting time: A field experiment. *Journal of Consumer Psychology* 12(3), 193–202 (2002)
6. Bech, S., Zacharov, N.: *Perceptual Audio Evaluation - Theory, Method and Application*. Wiley (July 2006)
7. Beerends, J.G., Hekstra, A.P., Hollier, M.P., Rix, A.W.: Perceptual Evaluation of Speech Quality (PESQ)-The New ITU Standard for End-to-End Speech Quality Assessment Part II-Psychoacoustic Model. *Journal of the Audio Engineering Society* 50(10), 765–778 (2002)
8. Belshe, M.: More Bandwidth does not Matter (much). Tech. rep., Google (2010)
9. Bhatti, N., Bouch, A., Kuchinsky, A.: Integrating User-Perceived quality into web server design. In: *9th International World Wide Web Conference*, pp. 1–16 (2000)
10. Bjorksten, M., Pohjola, O.P., Kilkki, K.: Applying user segmentation and estimating user value in techno-economic modeling. In: *6th Conference on Telecommunication Techno-Economics, CTTE 2007*, pp. 1–6 (2007)
11. Boev, A., Poikela, M., Gotchev, A., Aksay, A.: Modelling of the stereoscopic hvs. Tech. rep., Mobile3DTV Project, http://sp.cs.tut.fi/mobile3dtv/results/tech/D5.3_Mobile3DTV_v2.0.pdf
12. Bouch, A., Sasse, M.A., DeMeer, H.G.: Of packets and people: a user-centered approach to quality of service. In: *Proceedings of IWQoS 2000* (2000)
13. Bouch, A., Kuchinsky, A., Bhatti, N.: Quality is in the eye of the beholder: meeting users’ requirements for internet quality of service. In: *CHI 2000: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 297–304. ACM, New York (2000)
14. Van den Broeck, W., Jacobs, A., Staelens, N.: Integrating the everyday-life context in subjective video quality experiments. In: *2012 Fourth International Workshop on Quality of Multimedia Experience (QoMEX)*, pp. 19–24 (July 2012)

15. Brooks, P., Hestnes, B.: User measures of quality of experience: why being objective and quantitative is important. *IEEE Network* 24(2), 8–13 (2010)
16. Buchinger, S., Robitza, W., Hummelbrunner, P., Nezveda, M., Sack, M., Hlavacs, H.: Slider or glove? proposing an alternative quality rating methodology. In: *VPQM 2010* (2010), <http://eprints.cs.univie.ac.at/93/>
17. Callet, P.L., Möller, S.: Perkis (eds), A.: *Qualinet white paper on definitions of quality of experience* (June 2012)
18. Chen, K.T., Chang, C.J., Wu, C.C., Chang, Y.C., Lei, C.L.: Quadrant of euphoria: a crowdsourcing platform for QoE assessment. *IEEE Network* 24(2), 28–35 (2010)
19. Chen, W., Fournier, J., Barkowsky, M., Callet, P.L.: New requirements of subjective video quality assessment methodologies for 3dtv. In: *VPQM 2010* (2010)
20. Clark, A.: Modeling the effects of burst packet loss and recency on subjective voice quality. In: *Internet Telephony Workshop*, New York (April 2001), <http://www.cs.columbia.edu/~hgs/papers/iptel2001/21.pdf>
21. Collange, D., Costeux, J.L.: Passive estimation of quality of experience. *Journal of Universal Computer Science* 14(5), 625–641 (2008)
22. Crawley, E., Nair, R., Rajagopalan, B., Sandick, H.: RFC 2386: A Framework for QoS-based Routing in the Internet. Tech. rep., IETF (August 1998), <http://www.ietf.org/rfc/rfc2386.txt>
23. Cui, H., Biersack, E.: Trouble shooting interactive web sessions in a home environment, p. 25. *ACM Press* (2011), <http://64.238.147.56/citation.cfm?id=2018567.2018574&coll=DL&dl=GUIDE&CFID=103748494&CFTOKEN=26870697>
24. Daly, S.: The visible differences predictor: an algorithm for the assessment of image fidelity. In: *Digital images and Human Vision*, pp. 179–206. *MIT Press*, Cambridge (1993)
25. Dobrian, F., Sekar, V., Awan, A., Stoica, I., Joseph, D., Ganjam, A., Zhan, J., Zhang, H.: Understanding the impact of video quality on user engagement. In: *Proceedings of the ACM SIGCOMM 2011 Conference (SIGCOMM 2011)*, pp. 362–373. *ACM*, New York (2011), <http://doi.acm.org/10.1145/2018436.2018478>
26. Egger, S., Reichl, P., Hoßfeld, T., Schatz, R.: 'Time is Bandwidth'? Narrowing the Gap between Subjective Time Perception and Quality of Experience. In: *IEEE ICC 2012 - Communication QoS, Reliability and Modeling Symposium (ICC 2012 CQRM)*, Ottawa, Ontario, Canada (June 2012)
27. Egger, S., Schatz, R., Scherer, S.: It Takes Two to Tango - Assessing the Impact of Delay on Conversational Interactivity on Perceived Speech Quality. In: *INTERSPEECH*, pp. 1321–1324 (2010)
28. Egger, S., Schatz, R., Schoenenberg, K., Raake, A., Kubin, G.: Same but Different? - Using Speech Signal Features for Comparing Conversational VoIP Quality Studies. In: *IEEE ICC 2012 - Communication QoS, Reliability and Modeling Symposium (ICC 2012 CQRM)*, Ottawa, Ontario, Canada (June 2012)
29. Eisler, H.: Subjective duration and psychophysics. *Psychological Review* 82(6), 429–450 (1975)
30. Elert, G.: Wavelength of maximum human visual sensitivity, <http://hypertextbook.com/facts/2007/SusanZhao.shtml>
31. Fiedler, M.: EuroNGI deliverable D.WP.JRA.6.1.1: state of the art with regards to user perceived quality of service and quality feedback. Tech. rep. (May 2004), <http://eurongi.enst.fr>

32. Fiedler, M., Hofffeld, T., Tran-Gia, P.: A generic quantitative relationship between quality of experience and quality of service. *Netw. Mag. of Global Inter-
netwkg* 24, 36–41 (2010)
33. Garcia, M.N., Raake, A., List, P.: Towards content-related features for parametric video quality prediction of IPTV services, pp. 757–760 (2008)
34. Garcia, M.N., Raake, A.: Frame-layer packet-based parametric video quality model for encrypted video in IPTV services. In: *QoMEX 2011*, pp. 102–106 (2011)
35. Gomez, G., Sanchez, R.: End-to-End Quality of Service Over Cellular Networks: Data Services Performance and Optimization in 2g/3g. John Wiley (July 2005)
36. Gozdecki, J., Jajszczyk, A., Stankiewicz, R.: Quality of service terminology in IP networks. *IEEE Communications Magazine* 41(3), 153–159 (2003)
37. Grondin, S.: From physical time to the first and second moments of psychological time. *Psychological Bulletin* 127(1), 22–44 (2001)
38. Grondin, S.: Sensory modalities and temporal processing. In: Helfrich, H. (ed.) *Time and mind II: information processing perspectives*. Hogrefe & Huber (2003)
39. Grondin, S.: Timing and time perception: a review of recent behavioral and neuroscience findings and theoretical directions. *Attention Perception Psychophysics* 72(3), 561–582 (2010)
40. Hands, D., Wilkins, M.: A Study of the Impact of Network Loss and Burst Size on Video Streaming Quality and Acceptability. In: Díaz, M., Sénac, P. (eds.) *IDMS 1999*. LNCS, vol. 1718, pp. 45–57. Springer, Heidelberg (1999), <http://www.springerlink.com/content/21u2413r58534152/abstract/>
41. Hogarth, R.M.: *Judgement and choice: the psychology of decision*. J. Wiley (1980)
42. Hölldtke, K., Raake, A., Möller, S., Egger, S., Schatz, R., Rohrer, N.: How the Need for fast Interaction affects the Impact of Transmission Delay on the overall Quality Judgment. Tech. rep., ITU-T, Geneva, Switzerland (January 2011)
43. Hofffeld, T., Hock, D., Tran-Gia, P., Tutschku, K., Fiedler, M.: Testing the IQX Hypothesis for Exponential Interdependency between QoS and QoE of Voice Codecs iLBC and G.711. Tech. Rep. 442, University of Wuerzburg (March 2008)
44. Hofffeld, T., Schatz, R., Biedermann, S., Platzer, A., Egger, S., Fiedler, M.: The Memory Effect and Its Implications on Web QoE Modeling. In: *23rd International Teletraffic Congress (ITC 2011)*, San Francisco, USA (September 2011)
45. Hofffeld, T., Schatz, R., Seufert, M., Hirth, M., Zinner, T., Tran-Gia, P.: Quantification of YouTube QoE via Crowdsourcing. In: *IEEE International Workshop on Multimedia Quality of Experience - Modeling, Evaluation, and Directions (MQoE 2011)*, Dana Point, CA, USA (December 2011)
46. Hofffeld, T., Schatz, R., Varela, M., Timmerer, C.: Challenges of QoE Management for Cloud Applications. *IEEE Communications Magazine* (April 2012)
47. Hofffeld, T., Tran-Gia, P., Fiedler, M.: Quantification of quality of experience for edge-based applications. In: *20th International Teletraffic Congress (ITC 20)*, Ottawa, Canada (June 2007)
48. Hu, J., Wildfeuer, H.: Use of content complexity factors in video over ip quality monitoring. In: *2009 International Workshop on Quality of Multimedia Experience*, pp. 216–221 (2009)
49. Huynh-Thu, Q., Garcia, M., Speranza, F., Corriveau, P., Raake, A.: Study of rating scales for subjective quality assessment of High-Definition video. *IEEE Transactions on Broadcasting* 57(1), 1–14 (2011)
50. Ibarrola, E., Liberal, F., Taboada, I., Ortega, R.: Web qoe evaluation in multi-agent networks: Validation of itu-t g.1030. In: *ICAS 2009: Proceedings of the 2009 Fifth International Conference on Autonomic and Autonomous Systems*, pp. 289–294. IEEE Computer Society, Washington, DC (2009)

51. International Telecommunication Union: Handbook on Telephony, ITU-T (July 1992)
52. International Telecommunication Union: Terms and definitions related to quality of service and network performance including dependability. ITU-T Recommendation E.800 (August 1994)
53. International Telecommunication Union: End-user multimedia QoS categories. ITU-T Recommendation G.1010 (November 2001)
54. International Telecommunication Union: Analysis and interpretation of INMD voice-service measurements. ITU-T Recommendation P.562 (May 2004)
55. International Telecommunication Union: Estimating end-to-end performance in ip networks for data applications. ITU-T Recommendation G.1030 (November 2005)
56. International Telecommunication Union: Vocabulary and effects of transmission parameters on customer opinion of transmission quality, amendment 2. ITU-T Recommendation P.10/G.100 (2006)
57. International Telecommunication Union: H.264: Advanced video coding for generic audiovisual services. Series H: Audiovisual and Multimedia Systems; Infrastructure of audiovisual services - Coding of moving video (November 2007)
58. International Telecommunication Union: Subjective Evaluation of Conversational Quality. ITU-T Recommendation P.805 (July 2007)
59. Ito, Y., Tasaka, S.: Quantitative assessment of user-level QoS and its mapping. *IEEE Transactions on Multimedia* 7(3), 572–584 (2005)
60. ITU-T Study Group 2: Teletraffic Engineering Handbook. ITU (2005)
61. Janowski, L., Papir, Z.: Modeling subjective tests of quality of experience with a generalized linear model. In: *QoMEX 2009, First International Workshop on Quality of Multimedia Experience*, California, San Diego (July 2009)
62. Jekosch, U.: *Voice and Speech Quality Perception: Assessment and Evaluation*. Signals and Communication Technology. Springer (2005), <http://books.google.at/books?id=Ef31HiSzq1QC>
63. Kanumuri, S., Cosman, P., Reibman, A., Vaishampayan, V.: Modeling packet-loss visibility in MPEG-2 video. *IEEE Transactions on Multimedia* 8(2), 341–355 (2006)
64. Kawano, T., Yamagishi, K.: Performance evaluation of subjective quality assessment methods for stereoscopic video services, [ftp://vqeg.its.blrdoc.gov/Documents/VQEG_Hillsboro_Dec11/MeetingFiles/VQEG_3DTV_2011_037_Performance%20Evaluation%20of%203D%20Assessment%20Methods\(NTT\).doc](ftp://vqeg.its.blrdoc.gov/Documents/VQEG_Hillsboro_Dec11/MeetingFiles/VQEG_3DTV_2011_037_Performance%20Evaluation%20of%203D%20Assessment%20Methods(NTT).doc)
65. Khirman, S., Henriksen, P.: Relationship between Quality-of-Service and quality-of-experience for public internet service. In: *Proceedings of the 3rd Workshop on Passive and Active Measurement*, Fort Collins, Colorado, USA (March 2002)
66. Killeen, P.R., Weiss, N.A.: Optimal timing and the weber function. *Psychological Review* 94(4), 455–468 (1987)
67. Kim, D.S., Tarraf, A.: Anique+: A new american national standard for non-intrusive estimation of narrowband speech quality: Research articles. *Bell Lab. Tech. J.* 12(1), 221–236 (2007), <http://dx.doi.org/10.1002/bltj.v12:1>
68. King, A.: *Speed Up Your Site: Web Site Optimization*. New Riders, Indianapolis (2003)
69. Knoche, H., De Meer, H., Kirsh, D.: Utility curves: mean opinion scores considered biased. In: *1999 Seventh International Workshop on Quality of Service, IWQoS 1999*, pp. 12–14 (1999)

70. Knoche, H.O.: Quality of experience in digital mobile multimedia services (July 2011), <http://discovery.ucl.ac.uk/1322706/>
71. Kumar, V.: *Managing Customers for Profit: Strategies to Increase Profits and Build Loyalty*. Pearson Prentice Hall (2008)
72. Laghari, K., Crespi, N., Connelly, K.: Toward total quality of experience: A QoE model in a communication ecosystem. *IEEE Communications Magazine* 50(4), 58–65 (2012)
73. Malfait, L., Berger, J., Kastner, M.: P.563 - The ITU-T Standard for Single-Ended Speech Quality Assessment. *IEEE Transactions on Audio, Speech & Language Processing* 14(6), 1924–1934 (2006)
74. Mandryk, R.L., Inkpen, K.M., Calvert, T.W.: Using psychophysiological techniques to measure user experience with entertainment technologies. *Behaviour & IT* 25(2), 141–158 (2006)
75. Möller, S., Berger, J., Raake, A., Wältermann, M., Weiss, B.: A new dimension-based framework model for the quality of speech communication services. In: 2011 Third International Workshop on Quality of Multimedia Experience (QoMEX), pp. 107–112 (September 2011)
76. Möller, S., Berger, J., Raake, A., Wältermann, M., Weiss, B.: A new dimension-based framework model for the quality of speech communication services. In: 2011 Third International Workshop on Quality of Multimedia Experience (QoMEX), pp. 107–112 (September 2011)
77. Möller, S., Chan, W., Cote, N., Falk, T., Raake, A., Wältermann, M.: Speech quality estimation: Models and trends. *IEEE Signal Processing Magazine* 28(6), 18–28 (2011)
78. Möller, S.: *Assessment and Prediction of Speech Quality in Telecommunications*, 1st edn. Springer (August 2000)
79. Möller, S.: *Quality Engineering - Qualität kommunikationstechnischer Systeme*. Springer (2010)
80. Möller, S., Raake, A.: Telephone speech quality prediction: towards network planning and monitoring models for modern network scenarios. *Speech Communication* 38, 47–75 (2002), <http://portal.acm.org/citation.cfm?id=638078.638082>, ACM ID: 638082
81. Moor, K.D., Ketyko, I., Joseph, W., Deryckere, T., Marez, L.D., Martens, L., Verleye, G.: Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting. *Mobile Networks and Applications* 15(3), 378–391 (2010), <http://www.springerlink.com/index/10.1007/s11036-010-0223-0>
82. Moorsel, A.V.: Metrics for the internet age: Quality of experience and quality of business. Tech. rep., 5th Performability Workshop (2001), <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.3810>
83. Nielsen, J.: *Usability Engineering*. Morgan Kaufmann Publishers, San Francisco (1993)
84. Niida, S., Uemura, S., Nakamura, H.: Mobile services. *IEEE Vehicular Technology Magazine* 5(3), 61–67 (2010)
85. Nokia: Quality of experience (QoE) of mobile services - can it be measured and improved? (2004)
86. O’Shaughnessy, D.: *Speech communications: human and machine*, vol. 37. IEEE Press (2000), <http://www.ncbi.nlm.nih.gov/pubmed/11442089>
87. Olshefski, D., Nieh, J.: Understanding the management of client perceived response time. In: *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, pp. 240–251 (2006)

88. Osaka, N., Kakehi, K., Iai, S., Kitawaki, N.: A model for evaluating talker echo and sidetone in a telephone transmission network. *IEEE Transactions on Communications* 40(11), 1684–1692 (1992)
89. Pescapè, A., Salgarelli, L., Dimitropoulos, X. (eds.): TMA 2012. LNCS, vol. 7189. Springer, Heidelberg (2012)
90. Pine, B.J., Gilmore, J.H.: *The Experience Economy: Work is Theatre & Every Business a Stage*. Harvard Business Press (April 1999)
91. Pinson, M., Wolf, S.: Comparing subjective video quality testing methodologies. In: *SPIE Video Communications and Image Processing Conference*, pp. 8–11 (2003)
92. Pinson, M.H., Janowski, L., Pepion, R., Huynh-Thu, Q., Schmidmer, C., Coriveau, P., Younkin, A., Le Callet, P., Barkowsky, M., Ingram, W.: The influence of subjects and environment on audiovisual subjective tests: An international study. *IEEE Journal of Selected Topics in Signal Processing* 6(6), 640–651 (2012)
93. Pinson, M., Speranza, F.: Report on the validation of video quality models for high definition video content,
http://www.its.bldrdoc.gov/media/4212/vqeg_hdtv_final_report_version_2.0.zip
94. Pinson, M., Thorpe, L., Cermak, G.: Test plan for evaluation of video quality models for use with high definition tv content,
http://www.its.bldrdoc.gov/media/5871/vqeg_hdtv_testplan_v3_1.doc
95. Raake, A.: Short- and Long-Term Packet Loss Behavior: Towards Speech Quality Prediction for Arbitrary Loss Distributions. *IEEE Transactions on Audio, Speech, and Language Processing* 14(6), 1957–1968 (2006)
96. Raake, A., Garcia, M., Möller, S., Berger, J., Kling, F., List, P., Johann, J., Heidemann, C.: T-V-model: parameter-based prediction of IPTV quality. In: *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008*, pp. 1149–1152 (2008)
97. Raake, A.: *Speech Quality of VoIP: Assessment and Prediction*. John Wiley & Sons (2006)
98. Reichelt, S., Haussler, R., Fütterer, G., Leister, N.: Most 7690(0), 76900B–76900B–12 (2010)
99. Reichl, P., Egger, S., Schatz, R., D’Alconzo, A.: The Logarithmic Nature of QoE and the Role of the Weber-Fechner Law in QoE Assessment. In: *Proceedings of the 2010 IEEE International Conference on Communications*, pp. 1–5 (May 2010)
100. Reichl, P.: From charging for quality of service to charging for quality of experience. *Annales des Télécommunications* 65(3-4), 189–199 (2010)
101. Reichl, P., Tuffin, B., Schatz, R.: Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience. *Telecommunication Systems*, 1–14 (2011)
102. Richards, D.: Calculation of opinion scores for telephone connections. *Proceedings of the Institution of Electrical Engineers* 121(5), 313–323 (1974)
103. Rix, A.W., Beerends, J.G., Hollier, M.P., Hekstra, A.P.: Perceptual evaluation of speech quality (PESQ): the new ITU standard for end-to-end speech quality assessment - Part I: time-delay compensation. *Journal of the Audio Engineering Society* 50(10), 755–764 (2002)
104. Rubino, G.: Quantifying the quality of audio and video transmissions over the internet: the PSQA approach. In: *Design and Operations of Communication Networks: A Review of Wired and Wireless Modelling and Management Challenges*. Imperial College Press (2005)

105. Schatz, R., Egger, S.: Vienna Surfing - Assessing Mobile Broadband Quality in the Field. In: Taft, N., Wetherall, D. (eds.) Proceedings of the 1st ACM SIGCOMM Workshop on Measurements Up the STack (W-MUST). ACM (2011)
106. Schatz, R., Egger, S., Platzer, A.: Poor, Good Enough or Even Better? Bridging the Gap between Acceptability and QoE of Mobile Broadband Data Services. In: Proceedings of the 2011 IEEE International Conference on Communications, pp. 1–6 (June 2011)
107. Schatz, R., Egger, S., Platzer, A.: Poor, Good Enough or Even Better? Bridging the Gap between Acceptability and QoE of Mobile Broadband Data Services. In: International Conference on Communications (2011)
108. Seow, S.C.: Designing and Engineering Time: The Psychology of Time Perception in Software. Addison-Wesley Professional (2008)
109. Shaikh, J., Fiedler, M., Collange, D.: Quality of experience from user and network perspectives. *Annals of Telecommunications* 65, 47–57 (2010), <http://dx.doi.org/10.1007/s12243-009-0142-x>, 10.1007/s12243-009-0142-x
110. Siller, M., Woods, J.C.: QoS arbitration for improving the QoE in multimedia transmission. In: International Conference on Visual Information Engineering, VIE 2003, pp. 238–241 (2003)
111. da Silva, A.P.C., Varela, M., de Souza e Silva, E., Leão, R.M.M., Rubino, G.: Quality assessment of interactive voice applications. *Comput. Netw.* 52(6), 1179–1192 (2008)
112. Skadberg, Y.X., Kimmel, J.R.: Visitors' flow experience while browsing a Web site: its measurement, contributing factors and consequences. *Computers in Human Behavior* 20, 403–422 (2004)
113. Soldani, D., Li, M., Cuny, R.: QoS and QoE management in UMTS cellular systems. John Wiley and Sons (August 2006)
114. Staelens, N., Sedano, I., Barkowsky, M., Janowski, L., Brunnstrom, K., Callet, P.L.: Standardized toolchain and model development for video quality assessment - the mission of the joint effort group in vqeg. In: 2011 Third International Workshop on Quality of Multimedia Experience Quality of Multimedia Experience (QoMEX), Belgium, pp. 17–22 (September 2011)
115. Stankiewicz, R., Jajszczyk, A.: A survey of qoe assurance in converged networks. *Computer Networks* 55(7), 1459–1473 (2011)
116. Stevens, S.S.: On the Psychophysical Law. *Psychology Revue* 64(3), 153–181 (1957)
117. Strohmeier, D., Jumisko-Pyykk, S., Kunze, K.: Open profiling of quality: A mixed method approach to understanding multimodal quality perception. In: Advances in Multimedia 2010, pp. 1–28 (2010), <http://www.hindawi.com/journals/am/2010/658980/abs/>
118. Sun, L., Ifeachor, E.: Voice quality prediction models and their application in voip networks. *IEEE Transactions on Multimedia* 8(4), 809–820 (2006)
119. Takahashi, A.: Opinion model for estimating conversational quality of voip. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004, vol. 3, pp. 1072–1075 (May 2004)
120. Takahashi, A., Kurashima, A., Yoshino, H.: Objective assessment methodology for estimating conversational quality in voip. *IEEE Transactions on Audio, Speech, and Language Processing* 14(6), 1984–1993 (2006)
121. Thakolsri, S., Kellerer, W., Steinbach, E.G.: Qoe-based cross-layer optimization of wireless video with unperceivable temporal video quality fluctuation. In: ICC, pp. 1–6. IEEE (2011), <http://dblp.uni-trier.de/db/conf/icc/icc2011.html#ThakolsriKS11>

122. Varela, M.: Pseudo-subjective Quality Assessment of Multimedia Streams and its Applications in Control. PhD thesis, University of Rennes 1, France (2005)
123. Wältermann, M., Raake, A., Möller, S.: Quality Dimensions of Narrowband and Wideband Speech Transmission. *Acta Acustica united with Acustica* 96(6) (2010), <http://dx.doi.org/10.3813/AAA.918370>
124. Wang, Y.: Survey of objective video quality measurements. *Quality*, pp. 1–7 (2006)
125. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4), 600–612 (2004)
126. Wang, Z., Lu, L., Bovik, A.C.: Video quality assessment based on structural distortion measurement (February 2004)
127. Weber, E.H.: *De Pulsu, Resorptione, Auditu Et Tactu. Annotationes Anatomicae Et Physiologicae*. Koehler, Leipzig (1834)
128. Weinreich, H., Obendorf, H., Herder, E., Mayer, M.: Not quite the average: An empirical study of web use. *ACM Transactions on the Web (TWEB)* 2(1), 1–31 (2008)
129. Wilson, G.M., Sasse, M.A.: Do Users Always Know What’s Good For Them? Utilising Physiological Responses to Assess Media Quality. In: *The Proceedings of HCI 2000: People and Computers XIV - Usability or Else (HCI 2000)*, pp. 327–339. Springer (2000)
130. Winkler, S., Mohandas, P.: The evolution of video quality measurement: From PSNR to hybrid metrics. *IEEE Transactions on Broadcasting* 54(3), 660–668 (2008), <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4550731>
131. Winkler, S.: Issues in vision modeling for perceptual video quality assessment. *Signal Processing* 78(2), 231–252 (1999), <http://www.sciencedirect.com/science/article/pii/S0165168499000626>
132. Winkler, S.: Video quality measurement standards: current status and trends. In: *Proceedings of the 7th International Conference on Information, Communications and Signal Processing, ICICS 2009*, pp. 848–852. IEEE Press, Piscataway (2009)
133. Wolf, S., Pinson, M.: NTIA Report 02-392: Video Quality Measurement Techniques. Tech. rep., Institute for Telecommunication Sciences, <http://www.its.bldrdoc.gov/pub/ntia-rpt/02-392/>
134. Xue, J., Chen, C.W.: A study on perception of mobile video with surrounding contextual influences. In: *2012 Fourth International Workshop on Quality of Multimedia Experience (QoMEX)*, pp. 248–253 (July 2012)
135. Zepernick, H., Engelke, U.: Quality of experience of multimedia services: Past, Present, and Future. *ACM*, Lisbon (2011), <http://www.bth.se/fou/forskininfo.nsf/all/bb7dbb41bc492428c12578f7003d6bfb>
136. Zhang, J., Ansari, N.: On assuring end-to-end qoe in next generation networks: challenges and a possible solution. *IEEE Communications Magazine* 49(7), 185–191 (2011), <http://dblp.uni-trier.de/db/journals/cm/cm49.html#ZhangA11a>

Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience

Tobias Hößfeld¹, Raimund Schatz², Ernst Biersack³, and Louis Plissonneau⁴

¹ University of Würzburg, Institute of Computer Science, Germany
tobias.hossfeld@uni-wuerzburg.de

² Telecommunications Research Center Vienna (FTW), Vienna, Austria
schatz@ftw.at

³ Eurecom, Sophia Antipolis, France
erbi@eurecom.fr

⁴ Orange Labs, France
louis.plissonneau@orange.com

Abstract. This chapter investigates HTTP video streaming over the Internet for the YouTube platform. YouTube is used as concrete example and case study for video delivery over the Internet, since it is not only the most popular online video platform, but also generates a large share of traffic on today's Internet. We will describe the YouTube infrastructure as well as the underlying mechanisms for optimizing content delivery. Such mechanisms include server selection via DNS as well as application-layer traffic management. Furthermore, the impact of delivery via the Internet on the user experienced quality (QoE) of YouTube video streaming is quantified. In this context, different QoE monitoring approaches are qualitatively compared and evaluated in terms of the accuracy of QoE estimation.

1 Introduction

Quality of Experience (QoE) describes the user perception and satisfaction with application and service performance in communication networks, a topic that has gained increasing attention during the last years. Part of this growth of interest in QoE can be explained by increased competition amongst providers and operators, and by the risk that users churn as they become dissatisfied. However, many users face volatile network conditions, e.g. due to temporary over-utilization of shared network resources such as peering links. Such conditions may result in bad QoE. The focus of this book chapter is on Internet video delivery, since video streaming dominates global Internet traffic and exceeded half of global consumer Internet traffic at the end of 2011 [1]. We differentiate two different types of video content delivery over the Internet, (i) live video streaming with on-the-fly encoding, like IPTV, and (ii) streaming of pre-encoded video, so called Video-on-Demand (VoD). In this chapter, we focus on YouTube, the most popular VoD service in the Internet with more than two billion video streams daily.

The recent surge in popularity of Internet video requires a considerable investment by the operators of these services in order to be able to satisfy the demand. The **delivery infrastructure** (Section 2) is generally distributed all over the world and comprises of tens of different sites. A user must be automatically directed to a nearby site (which is

the role of DNS) and must be redirected in case the video is not available on this particular site or if the servers of this site are overloaded. Thus, a cache selection mechanism is implemented within the delivery infrastructure. Today, the delivery of the video data is typically performed via TCP, which is a reliable transport protocol that performs error recovery and congestion control. When using TCP, the transmission can be subject to considerable delay jitter and throughput variations and the client needs to preload a play out buffer before starting the video playback. Various transmission strategies from the server to the client are possible such as client pull or server push mode. Also the size of the video blocks transmitted can vary from 64 Kbytes to a few Mbytes. This is referred to as application-layer traffic management.

For an Internet Service Provider (ISP) providing connectivity to the end user, it is thus important to understand the relationship between Quality of Experience (QoE) of a service and the performance characteristics of the service provisioning through networks, resulting into a so-called **QoE model** (Section 3). In the context of network provisioning, QoE also opens the possibility to save resources by proper QoE management, as it is not economic to invest in better Quality of Service (QoS) for maintaining the same level of QoE. For example, reserving a bandwidth of 16 Mbps for delivering a video stream that has a video bit rate of only 300 Kbps unnecessarily consumes ISP's resources without improving QoE.

To implement a QoE management, the ISP must identify and monitor the traffic in its network that results from that service. These measurement data is used for estimating the QoE by means of an appropriate QoE model. Since YouTube video streams do not change encoding parameters during playback and packet loss artifacts do not occur (due to the use of TCP), the QoE of YouTube is primarily determined by stalling effects on application layer as opposed to image degradation in UDP-based video streaming.

In this book chapter, we investigate **QoE monitoring for YouTube video streaming**. To this end, Section 2 provides the necessary technical background and fundamentals in terms of the delivery infrastructure, cache selection, and traffic management mechanisms that enable such a service. Section 3 then investigates the end-user perspective by identifying the key influence factors for YouTube QoE and developing a model that maps application-layer QoS to QoE. Finally, Section 4 brings these elements together to present various YouTube QoE monitoring approaches both, at the application-layer and the network-layer. The approaches at these two layers are fundamentally different: Application-layer monitoring requires to change the end user application or to install additional monitoring software, but will lead to exact results since performance is directly measured at the user terminal where QoE impairments become directly perceivable. On the other hand, a highly scalable, valid, and robust QoE monitoring approach (including a stalling detector as basic QoE indicator) from measurements within the network is needed by ISPs to detect any problems in the network or to manage the traffic accordingly to overcome networking problems. To this end, several network-layer monitoring approaches of YouTube QoE, as well as an evaluation on its accuracy (compared to application-level monitoring) and implementation prospects are highlighted.

2 Delivery Infrastructure, Cache Selection and Application-Layer Traffic Management

Recent studies show that videos streaming sites represent about half of the Internet data volume, both for mobile access [2] and for fixed access [3]. Moreover, video streaming today produces a traffic volume that is more than double the one due to peer-to-peer. In the case of peer-to-peer the server resources to satisfy the demand come from the peer themselves, while in the case of video streaming a content distribution network must be built that consists of geo-distributed servers and caches. It is therefore instructive to study in detail the organization of a video streaming service and its evolution over time. In the following, we will present the design principles of the YouTube delivery architecture and its performance.

2.1 Basic Facts about YouTube

YouTube, which was created in 2005, allows users to upload and share video content. Its success was immediate, resulting in spectacular growth ever since. For instance, the *number of videos viewed per day* has increased from around 200 Million in 2007 to more than 4 Billion in 2012 [4]. Since YouTube was acquired in late 2007 by Google, its infrastructure has been in constant evolution and the delivery architecture that initially used third party content distribution network services is now fully operated and managed by Google. Not much about YouTube has been disclosed by Google itself [4–6]. However, in the last couple of years YouTube has been extensively investigated [7–12] by academia via active and/or passive measurements, which are often carried out from multiple vantage points. While such measurements can reveal certain aspects of YouTube, many details are still unknown. Our description of YouTube is based on the results published in literature and on recent studies of YouTube carried out by ourselves. Describing a system like YouTube that constantly evolves is challenging. However, we believe that the underlying *design principles* will most likely stay the same for some time. We sometimes give real figure to indicate the size of YouTube with the aim to give an idea of the order of magnitude and to provide a reference point for future comparison.

The number of videos hosted by YouTube was estimated in mid-2011 [13] to be about 500 million, which represents about 5 PetaBytes of data considering an average size of 10 MBytes per video. Taking into account replication of videos and multiple formats, this makes a total of about 50 PetaBytes of data. In 2012, an average of *one hour of video was uploaded every second*⁰, which is a three-fold increase as compared to 2009/ The number of videos downloaded per day has been evaluated in 2011 to be between 1.7 and 4.6 Billion representing a 50% increase over the previous year, which results in tens of PetaBytes of traffic per day. While YouTube originally started its service in the USA, it has become truly international in the meantime with caches in many countries. Today only 25% of the views are generated in the USA, followed by countries such as UK, Japan, Germany or Brazil, which each generate between 3–7% of all the views [5]. Geographic request locality is high, with around two thirds of the views per

video coming from a single region, where a region represents a country or another political, geographic or linguistic entity [5]. Request locality also varies between countries and is highest in Japan and Brazil.

In the following, we focus on the process of downloading the videos. We are going to review the major steps in accessing a YouTube video before we describe the server infrastructure and explain cache selection and cache redirections, which are used to balance the load among caches.

When a user uploads a video to YouTube, the video is stored on a server in one of Google **backend data centers**. YouTube supports multiple video formats. Each video may be transcoded in all the different formats, which can happen pro-actively or on the fly. As we will see in the following, a user that requests a YouTube video will never directly interact with the servers in backend data centers. Instead, the videos will be delivered to the users from so called **caches**.

2.2 Steps in YouTube Video Download

Watching a video on YouTube involves a different set of servers. Initially, the embedding Web page is delivered through front end YouTube web servers, whereas the video content is itself delivered by YouTube video cache servers. YouTube currently supports two containers for video streaming, Flash and HTML5 [14]. At the time of writing, the adoption of HTML5 for YouTube playback on PCs is still in an early phase, and almost all browsers use Flash technology as the default to play the YouTube videos [10]. When the container is Flash, a dedicated Shockwave Flash player must be downloaded to control the Flash plugin in the browser.

Simplified Steps in Accessing a YouTube Video. As shown in Figure 1, the process of accessing a YouTube video can be summarized as (numbers correspond to the graph):

- (1) The user requests a video on the YouTube webpage: `http://www.youtube.com/watch?v=videoID` and gets to the Web server that delivers the YouTube HTML page;
- (2) After downloading the embedding HTML web page, the other contents are requested in particular the Shockwave Flash Player (embedded in a HTML object that contains the video parameters);
- (3) The actual video content is requested from a cache server (`lscache` server); if this cache is over-loaded, it sends a redirect (HTTP 302) message to the client indicating another cache server;
- (4) The client sends a request the other cache server (`tccache` server) for the video, and the FLV file is delivered to the client while being played in the Flash player (Progressive Download).

The details of the redirections depend on the load of the cache servers and are explained in the following.

We now focus on the video content delivery, and more specifically on the architecture and the interaction with the cache server infrastructure.

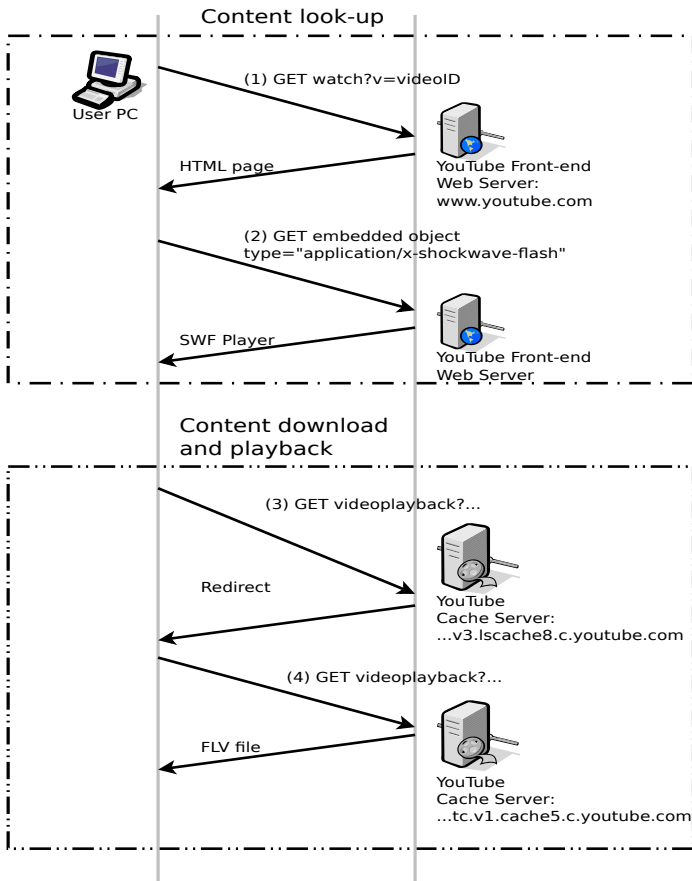


Fig. 1. Schema of YouTube page download

YouTube Cache Server Infrastructure. The users receive the video they request from a cache node. Individual **cache nodes** are organized in cache clusters, with all the machines of a **cache cluster** being co-located. The number of machines per cache cluster is highly variable and depends, among others, on the demand for service issued in the region where the cluster is located and also on the available physical space to host the cache nodes. Each cache node as of 2011 has a 10 Gb/sec network access and 78 TByte of disk storage [4].

The various cache clusters are organized in a three tier hierarchy. The global infrastructure of the YouTube caches has been revealed by Adhikari *et al.* [7] in 2011. They used the distributed infrastructure of the PlanetLab network to request thousands of videos from different vantage points in the world, which allowed to reverse engineer the cache infrastructure and the cache selection policies. We complement their findings with our own active measurements [15] undertaken in 2011 and 2012 from France. Our analysis focuses on residential Internet access and reveals the techniques applied by

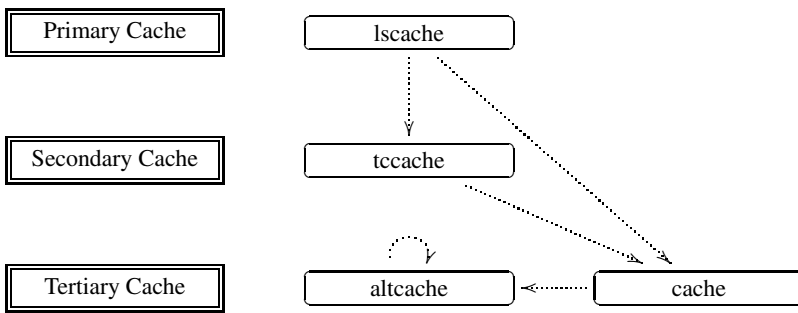


Fig. 2. Organization of the YouTube Caches; dashed lines indicate possible redirections

Google to deliver the videos to residential customers. Since our machines were connected to the Internet through *different ISPs*, we were able to observe differences in treatment of customers coming from different ISPs.

YouTube has a three tier caching infrastructure that comprises of four different *logical* namespaces as shown in Figure 2. The machines allocated to the different cache clusters are identified via particular naming conventions. We recall here the main findings of [7] on the YouTube cache clusters. As of 2011 there are:

- 38 primary cache clusters with about 5000 unique IP addresses corresponding to the `lscache` namespace;
- 8 secondary cache clusters corresponding to the `tccache` namespaces with about 650 IP addresses;
- 5 tertiary caches clusters corresponding to the `cache` and `altcache` namespaces with about 300 IP addresses.

All these cache clusters are located in a total of 47 different locations distributed over four continents; there is no cache cluster in Africa. About ten primary cache clusters are co-located inside ISPs and the IP addresses of these cache nodes in these clusters are not part of the address space managed by Google but part of the ISPs address space. Since we have $38 + 8 + 5 = 51$ cache clusters but only 47 different locations, some cache clusters belonging to different levels of the hierarchy must be at the same physical location (*i.e.* some primary and secondary caches are co-located).

For the caches in each cache cluster, a particular *logical* naming structure is applied.

- Each primary cache cluster has a total of 192 logical caches corresponding to the `lscache` namespace, which looks as follows: `city_code.v[1-24].lscache[1-8].c.youtube.com`. As `city_code` the three letter code for the airport closest to that cache cluster is used.
- There are also 192 logical caches in each secondary cache cluster, corresponding to the `tccache` namespace, which are named as follows `tc.v[1-24].cache[1-8].c.youtube.com`.
- Each tertiary cache cluster has 64 logical caches corresponding to `cache` and `altcache` namespaces.

Introducing these logical name spaces has the following advantages:

- Each video ID can be deterministically mapped via consistent hashing onto a unique logical name in the `lscache` namespace, which makes it easy to decide for each cache what portion of the videos it is responsible to serve.
- There is a one-to-one mapping between the `lscache` and `tccache` namespace.
- The logical naming is the same for each cache cluster and it is completely independent of the number of *real cache* nodes in a particular cache cluster.
- It is the responsibility of DNS to map logical cache names onto the IP addresses of real cache nodes. In [7], each of the logical names from the `lscache` namespace is mapped to more than 75 different IP addresses distributed over the 38 primary cache clusters.

YouTube Datacenter Sizes. We have carried out active measurements in France [15], using simultaneously nine different Internet accesses (7 ADSL and 2 Fiber) to request videos during sessions that lasted for 2 days each. All these accesses are in the same physical location and the access rates for all the ADSL accesses are the same.

The YouTube videos crawled during these measurements were served by two datacenters: one in Paris (`par`), the other in Amsterdam (`ams`). In Tab. 1 we show the number of IP addresses seen for each datacenter and match each IP address with its corresponding `lscache` namespace.

Tab. 1 gives an idea of the size of a cache cluster and also shows the evolution of these cache clusters over time:

- The Amsterdam cache cluster increased its size by 50% within a few months; for this cache cluster there are more IP addresses than distinct `lscache` names, which means that a single `lscache` name will be mapped onto several IP address.
- The Paris cache cluster re-organized the distribution of its IP addresses into *two distinct* logical `lscache` namespaces. For this cache cluster there are fewer IP addresses than distinct `lscache` names, which means that several `lscache` names will be mapped on the same IP address.

In Figure 2, we also show the dynamics of redirections inside the YouTube cache layers. Each cache layer can redirect to the next cache level and the tertiary cache layer can be accessed through redirection out of any layer (including itself). We will explain this in detail in the next section on cache selection.

2.3 YouTube Cache Selection

YouTube cache selection is quite sophisticated and tries to:

- Satisfy users by selecting a *nearby* cache cluster and
- Perform internal redirection to another cache cluster to perform load balancing among cache clusters.

The choice of a close-by cache cluster (in terms of RTT) is typically done through DNS resolution. DNS is used for coarse grained load balancing, with a TTL of five minutes. Before getting into the process of cache server selection through redirections, we first study the selection of first cache server.

Table 1. YouTube Datacenters sizes according to the Number of IP addresses seen for crawls of all ISPs on each URL Regexp

(a) September 2011

URL Regexp	# IPs
o-o.preferred.par08s01.v[1-24].lscache[1-8].c.youtube.com	160 [†]
o-o.preferred.par08s05.v[1-24].lscache[1-8].c.youtube.com	160 [†]
o-o.preferred.ams03g05.v[1-24].lscache[1-8].c.youtube.com	328
o-o.preferred.ISP-par1.v[1-24].lscache[1-8].c.youtube.com	98

[†] these two sets of 160 IP addresses are identical

(b) December 2011

URL Regexp	# IPs
o-o.preferred.par08s01.v[1-24].lscache[1-8].c.youtube.com	80 [‡]
o-o.preferred.par08s05.v[1-24].lscache[1-8].c.youtube.com	80 [‡]
o-o.preferred.ams03g05.v[1-24].lscache[1-8].c.youtube.com	494
o-o.preferred.ISP-par1.v[1-24].lscache[1-8].c.youtube.com	130

[‡] these two sets of 80 IP addresses are *distinct*

Choice of First Cache Server. With our active measurement carried out across different ISPs in France [15] we also wanted to investigate if clients from different ISPs get directed to the same cache cluster or not. We only focus on the first cache server returned and do not take into account redirections. All the accesses are located in the same place with the same access rate for ADSL. The city codes are par and ams for Paris and Amsterdam respectively.

We see from Tab. 2, cache cluster used to serve clients clearly depends on the ISP. Here are the main findings (cf. Tab. 2):

- ISP B has all its `lscache` names pointing to one cache site (`par08s01`) in Paris;
- ISP N has all its `lscache` names pointing to the Paris cache site, but with two different logical name spaces (`par08s01` and `par08s05`);
- ISP O has dedicated `lscache` names carrying the IPS name (`ISP-par1`). Also, these names get resolved to IP addresses that belong to a specific AS (36040), which is different from the Google or YouTube ASes.
- ISPs S and F are directed to both cache clusters in Paris or Amsterdam with different proportions: about 2/3 to Amsterdam for ISP S and 10% for ISP F.

These results highlight that there is a customization done for each ISP for reasons only known to Google.

The network impact of the cache cluster location on the ping time is found to be very low. For example, the minimum ping time from our lab in France to the Paris cache nodes is of 23.8ms and of 28ms to Amsterdam (because of relatively small distance between the two cities). However, the main point is that the cache cluster selected in

Table 2. Number of Videos for each ISP according to Regexp on lscache names for a controlled crawl in December 2011

URL Regexp	ISP								
	A [¶]	B [¶]	B [§]	F-1 [¶]	F-2 [¶]	N [§]	O [¶]	S-1 [¶]	S-2 [¶]
par08s01.v[1-24].lscache[1-8]	0	2676	2677	0	0	1890	0	1967	1528
par08s05.v[1-24].lscache[1-8]	1636	0	0	952	2425	799	0	0	0
ams03g05.v[1-24].lscache[1-8]	150	0	0	0	206	0	0	3033	2488
<i>ISP-par1.v[1-24].lscache[1-8]</i>	0	0	0	0	0	0	2591	0	0

[¶] ADSL access

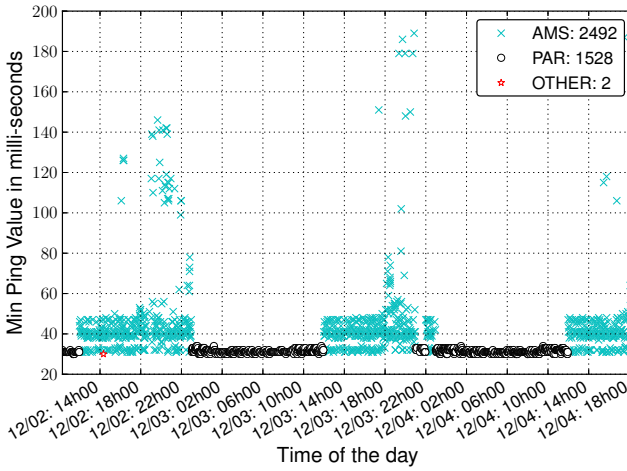
[§] Fiber access

not necessarily the geographically closest one and that the choice of the preferred cache cluster depends on the time of day as shown in Figure 3 and on the ISP the client is connected. Moreover, even if the minimum ping values for both cache clusters are about the same, the cross traffic on the path from France to Amsterdam can increase the ping value to values as high as 200 ms. Indeed, Figure 3 shows a large variance in ping times towards Amsterdam cache nodes. The most striking point is that the switch from one datacenter to another is done at a specific time every day, and this time is specific to each ISP.

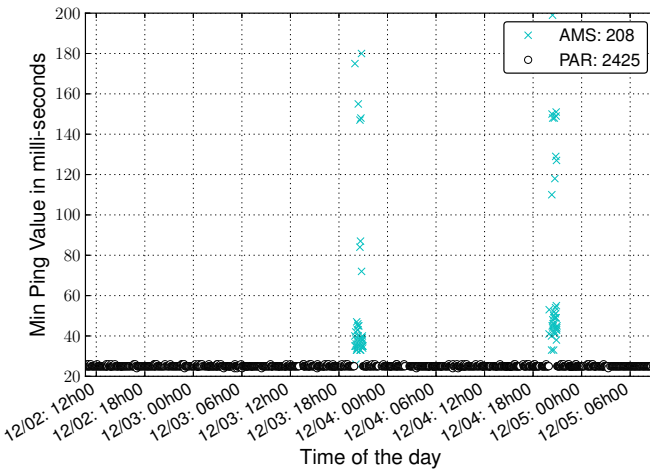
We have made the same observation in [15] for a different experiment with clients located in Kansas, USA, who were served from a variety of different cache clusters located anywhere in the US. In Figure 4, we present a partial map of USA with the location of the most prevalent cache clusters seen in this crawl. The symbols have the following meaning:

- The pin mark is the place from where the crawls are performed: Kansas City;
- Each circle corresponds to a YouTube cache site;
- The diameter of each circle represents the number of videos served by this cache site;
- The color of each circle represents the distance (ping time) towards the cache site: green for ping time lower than 60 ms, blue for ping time between 60 and 200 ms, and red for ping time larger than 200 ms.

Note that we do not show the San Jose and Los Angeles cache sites in the figure, which receive 11% and 3% respectively of the video requests. There are four more cache sites, which are located in Kansas-City, Lawrence, Chicago and New-York that receive a small fraction of all requests. The details can be found in [15]. We clearly see that the distance is not the primary criterion for the choice of the cache site: the most frequently used cache site is in Washington DC, even though it is much further away than the Dallas cache site.



(a) S-2



(b) F-2

Fig. 3. Ping time in milliseconds from our controlled lab towards two cache sites observed in a controlled crawl in December 2011

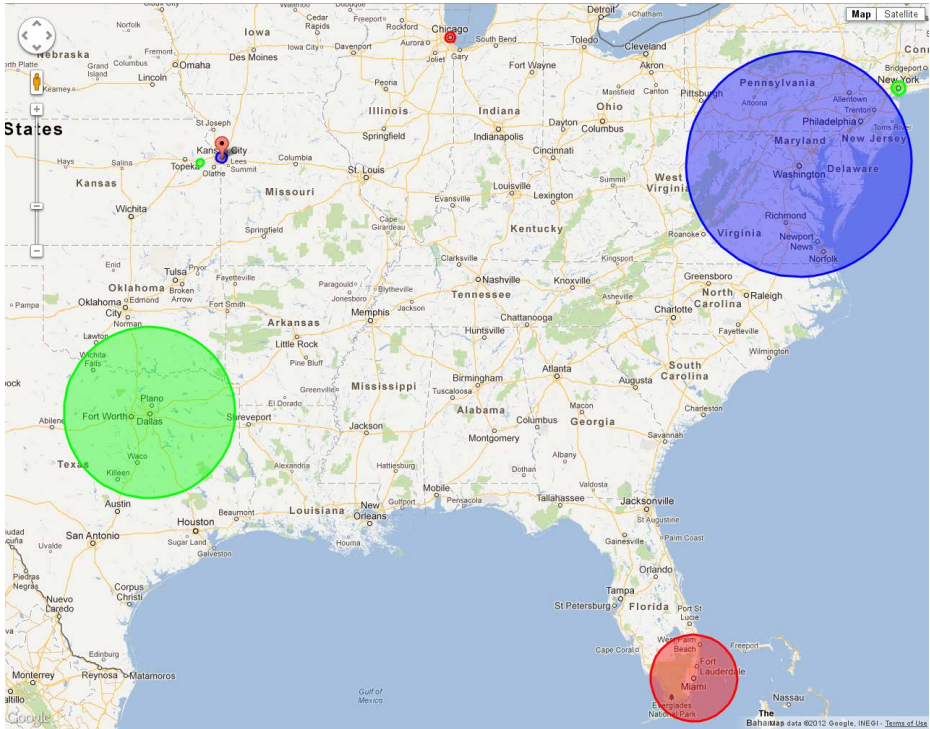


Fig. 4. Map indicates the cache site locations, number of requests served by each cache site (diameter of the circle), and distance (circle color: green for ping ≤ 60 ms, blue for ping ≥ 60 ms and ≤ 200 ms, and red for ping ≥ 200 ms) of the YouTube cache sites in the crawls that originate from Kansas City (pin mark).

Cache Selection through Redirections. We have already seen that load balancing on YouTube cache servers can be done through DNS resolution: this process is *centralized* at the YouTube authoritative DNS servers. Load-balancing can also be done directly at cache server level. In this case, the cache server receiving the request can relay the request to another cache server via HTTP redirect message at application level. So YouTube can use both *centralized* and *decentralized* processes to balance the requests on its cache servers.

Cache hit. If the video is hot and there are copies at the primary caches, then a logical cache node (`lscache` namespace) in the *primary cache* is chosen. If there is no redirection, a machine from a cache cluster serves the video.

If the primary cache cluster selected is overloaded, a redirection to a *secondary cache cluster* (`ccache` namespace) occurs. The secondary cache can serve the video or redirect it to a *tertiary cache site* (`cache` namespace) for load-balancing purposes.

Again, in the tertiary cache cluster, the cache server can deliver the video, or perform another redirection. A redirection from a tertiary cache site will remain *inside*

the tertiary cache level and occur towards a cluster from the `altcache` namespace. A machine in this namespace now serves the video or redirects it inside the same namespace in case of overload. Very rarely, several redirections occur inside the `altcache` namespace, with the total number of redirections being limited to 9. For further details see [7].

Cache Miss. If the video is cold and there are *no* copies at the primary caches, then the request will be most likely redirected from the first level cache to a third level cache. The third level cache will fetch the video from the backend data server, cache it, and deliver the video to the client. It is quite common, as we will see in the next section, that users do not watch the entire video. Therefore, all videos are broken into chunks (of 2 MBytes) and the cache will continue to retrieve from the backend servers new chunks of a video as long as the user keeps viewing that video. Note that despite all the efforts of the engineering team of Google, the cache miss rate remains steadily at about 10% [4].

YouTube Redirection Process

DNS Level Redirections. YouTube uses HTTP redirections to balance the load among its caches. As shown in Figure 2, the redirections usually direct the video request from a cache layer to the next one. Using traces from a European ISP, Torres *et al.* [12] observed that as the total number of requests kept increasing, the percentage of requests handled by the closest cache cluster located inside that ISP decreased from 80% to about 30%. In this case, DNS request resolution will direct clients to more distant but less loaded cache clusters.

Impact of Redirections on Performance. Each redirection involves:

1. DNS query to resolve the hostname of the next cache node,
2. Opening of a new TCP connection,
3. Issuing a new video query.

In case of redirections, the final cache node serving the video will most likely not be the closest one in terms of RTT, which has been observed in [12] for *the most popular videos of the day*.

The impact of redirection on the time until the first MByte is downloaded (referred to as video initialization time) has also been studied in [7]. The video initialization time is on average 33% higher if the video has been fetched through redirections. The fraction of sessions that have been redirected is evaluated in [10]: between 10% and 30% of all sessions are redirected at least once. The impact of redirections on the startup delay can also be important [10]:

- Without redirections, delays are in the order of milliseconds;
- With redirections, delay can increase by orders of magnitude, up to 10 seconds.

2.4 Application-Layer Traffic Management

YouTube videos are requested using HTTP over TCP. TCP is a transport protocol that assures reliable transfer by retransmitting lost data packets and performs congestion control to avoid overloading the network. Both error control and congestion control of TCP may result in high delay jitter.

The delivery strategy of YouTube videos has been studied in great detail by Rao *et al.* [14]. The authors show that the delivery strategy depends on the video container (Flash, Flash High Definition, or HTML5), the type of client device (PC or mobile devices such as smart phones or iPad), and the type of browser (Internet Explorer, Chrome, or Firefox). The delivery strategy needs to reconcile a number of potentially conflicting goals such as:

- Smooth playout during the entire duration of a viewing session;
- Efficient use of the server resources such as disk I/O and timer management;
- Avoid to transmit too much data in advance of consumption in order to (i) reduce the amount of buffering at the client, which is particularly relevant in the case of mobile devices and to (ii) reduce the waste of network and server resources by sending data that are never used.

Finamore *et al.* [10] observed that 60% of the videos requested were watched for less than 20% of their total duration, resulting in an un-necessary transfer of 25–39% of the data. As we shall see in the following section, the impact of playback degradation is a primary factor in the video transfer interruption.

As the video transfer is done via HTTP over TCP, there is not guarantee that the data can be delivered to the client at the rate at least as high as the one at which they are consumed. The details of the transfer have been studied in [14], whose findings we summarize in the following: To increase the likelihood of a smooth playback, YouTube performs aggressive buffering when a video is requested. Initially, during a **startup** phase, the server sends as fast as possible to fill up the initial client playout buffer. This playout buffer contains about 40 seconds with Flash, and 10–15 MBytes with HTML5 with Internet Explorer as a browser, which is typically much more than 40 seconds worth of video. Once the initial buffer has been filled, two other strategies are used by the cache server:

- keeps sending as fast as possible, until entire video is transmitted;
- limits the rate of the transfer alternating between on-off cycles with a fixed period. During an on cycle, a fixed size block of video data is transmitted.

We limit our description to the case of streaming a video to a PC with Flash as container, and refer to the original paper [14] for more details.

Streaming the video to a PC has been the most extensively studied [6, 16]. In this case, the server streaming strategy is independent of the browser: When the startup phase is terminated, the cache server sends blocks of 64 KBytes at a frequency that allows to achieve an average transmission rate of 1.25 times the video encoding rate. As has been first observed by Alock and Nelson [16], injecting bursts of 64 KBytes means sending 45 maximum size TCP segments back-to-back into the network. Such large packet bursts will accumulate in the buffer of the bottleneck link and (i) cause

delay spikes that may disrupt other latency sensitive application and (ii) inflict loss on the bursty YouTube flow itself. In response to these problems, Google engineers have recently proposed a modification to the server side sending policy that controls the amount of packets that can be injected back-to-back in order to limit the size of the packet bursts. For details of the new sender algorithm and its impact on packet loss and burst size see [6].

In this section we have provided the technical basis to understand YouTube content delivery over the Internet. Next, we investigate what influences the QoE experienced by the user. In particular, problems in the network may lead to stalling and QoE degradations. Therefore, we have to identify the key factors that influence YouTube QoE by means of subjective measurements and build an appropriate model, which can be used for QoE monitoring later on.

3 QoE of YouTube Video Streaming

User perceived quality of video streaming applications in the Internet is influenced by a variety of factors. As a common denominator, four different categories of influence factors [17, 18] are distinguished, which are influence factors on context, user, system, and content level.

- The **context level** considers aspects like the environment where the user is consuming the service, the social and cultural background, or the purpose of using the service like time killing or information retrieval.
- The **user level** includes psychological factors like expectations of the user, memory and recency effects, or the usage history of the application.
- The technical influence factors are abstracted on the **system level**. They cover influences of the transmission network, the devices and screens, but also of the implementation of the application itself like video buffering strategies.
- For video delivery, the **content level** addresses the video codec, format, resolution, but also duration, contents of the video, type of video and its motion patterns.

In this section, a simple QoE model for YouTube is presented whose primary focus is its application for QoE monitoring (within the network or at the edge of the network). Therefore, we take a closer look at objectively measurable influence factors, especially on the system and content level. For this purpose, subjective user studies are designed that take into account these influence factors; in particular, we utilize crowdsourcing to have a large pool of human subjects to conduct the tests (Section 3.1). The crowdsourcing environment also allows analyzing influence factors on user level and context level. After analyzing the user ratings that are the key influence factors on YouTube QoE (Section 3.2), simple QoE models and its corresponding mapping functions between those influence factors and the YouTube QoE can be derived (Section 3.3). For illustration, Figure 5 sketches the methodology from subjective user studies to QoE models for QoE monitoring.

3.1 Subjective User Studies for Quantifying YouTube QoE

Subjective user studies are the basis to quantify the YouTube QoE and to model the impact of influence factors on context, user, system, and content level. Therefore, realistic

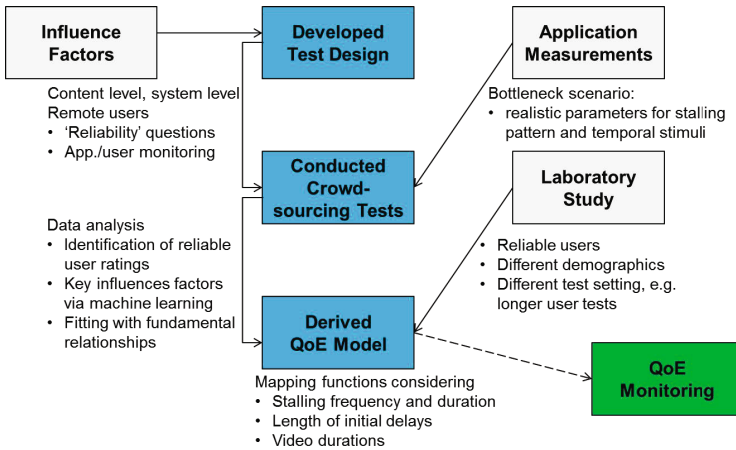


Fig. 5. Methodology applied from subjective user studies towards QoE models for QoE monitoring [19, 20]

test scenarios will be defined that consider typical video clips and stalling patterns. The general test methodology developed in [21] allows researchers to conduct subjective user tests about YouTube QoE by means of crowdsourcing. Further experiments were conducted in a laboratory environment to double-check the test results and to exclude the influence of the actual test setting and implementation.

Realistic Test Scenarios: Typical Stalling Patterns and Video Clips. The main goal of the experiments is to quantify the impact of system level influence factors, in particular network impairments on QoE. For YouTube video streaming, network impairments result into related *stalling patterns*. Stalling events during the video playout are caused by rebuffering of the video due to an insufficient content delivery rate. For example, if the video bit rate is larger than the available network or server data rate, the video buffer will emptied at some point in time and then the video freezes until the video buffer is filled again. As a consequence, the YouTube user has to wait until the video restarts playing. Furthermore, the user perceived quality suffers from *initial delays* before the YouTube video playout starts, since the player fills up the video buffer before the video playout. In general, the shift from unreliable media streaming to reliable HTTP over TCP streaming makes waiting times one of the key QoE influence factors in the domain of web-based video streaming. In the subjective user studies, these stalling patterns and also initial delays are simulated and then the user is asked about her user perceived quality – in presence of these waiting times.

To obtain realistic stalling patterns, the relationship between network QoS and stalling events must be derived, which is not trivial due to the application-layer traffic management by YouTube (see Section 2.4). In case of a bottleneck with a fixed network data rate, periodic stalling patterns occur [19], i.e., every Δt seconds a stalling event of almost fixed length L takes place. An illustration of the YouTube video buffer evolution in case of a bottleneck is depicted in Figure 6. As soon as the first threshold is exceeded,

the video playback starts. However, if the video bit rate is larger than the network data rate (which is here the case due to the bottleneck), the video buffer is emptied faster than the network can deliver video data. As soon as the video buffer falls below a certain threshold [19], the video stalls.

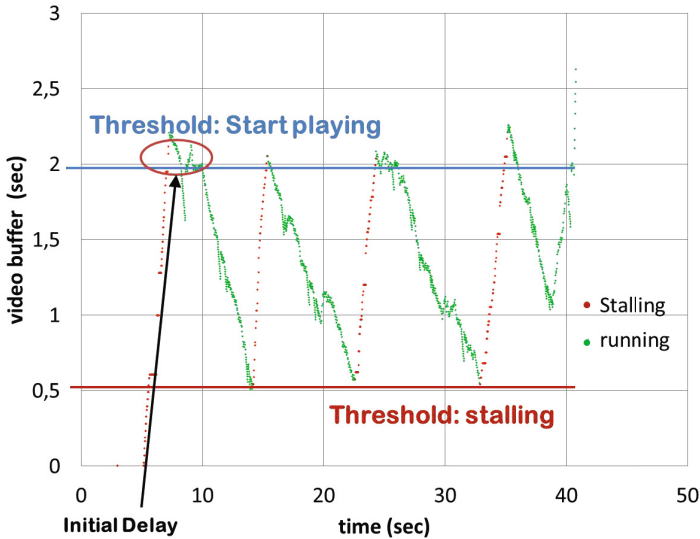


Fig. 6. Implementation of the YouTube video player [19]

On a content level, typical YouTube videos of various content classes like news, sports, music clips, cartoons, etc. were used in the tests. Thereby, the video clips had different resolutions, motion patterns and video codec settings. To reduce the state space of parameters to be tested, only 30 s and 60 s long videos are considered in the test results. On context level, realistic desktop settings are considered. Thus, the video experience in the test should be as similar as possible to a visit of the real YouTube website and the application should run on the users' default web browser.

Crowdsourcing QoE Tests. To conduct subjective user studies for YouTube QoE, crowdsourcing seems to be an appropriate approach. Crowdsourcing means to outsource a task (like video quality testing) to a large, anonymous crowd of users in the form of an open call. Crowdsourcing platforms in the Internet, like Amazon Mechanical Turk or Microworkers, offer access to a large number of geographically widespread users in the Internet and distribute the work submitted by an employer among the users. With crowdsourcing, subjective user studies can be *efficiently conducted at low cost* with an adequate number of users in order to obtain statistically significant QoE scores. In addition, the desktop-PC based setting of crowdsourcing provides a *highly realistic context* for usage scenarios like online video consumption.

However, the *reliability of results* cannot be taken for granted due to the anonymity and remoteness of participants: some subjects may submit incorrect results in order to maximize their income by completing as many tasks as possible; others just may not work correctly due to lack of supervision. To assure the quality of these QoE tests and identify unreliable user ratings, different task design methods are proposed in [21] like including content questions about the videos evaluated or consistency questions. For example, the user is asked about his origin country in the beginning and about his origin continent at the end of the test. The ratings of the participant are disregarded, if not all answers of the test questions are consistent. Furthermore, application-layer monitoring helps to identify reliably conducted tests. E.g. we monitored browser events in order to measure the focus time, which is the time interval during which the browser focus is on the website belonging to the user test. In addition to the crowdsourcing user studies, time-consuming laboratory studies were conducted to double-check the test results. The laboratory studies are described later in this section.

To have a realistic test scenario, the video experience in the test should mimic a visit of the real YouTube website. To this end, an instance of the YouTube Chromeless Player was embedded into dynamically generated web pages. With JavaScript commands the video stream can be paused, a feature we used to simulate stalling and initial delays. In addition, the JavaScript API allows monitoring the player and the buffer status, i.e. to monitor stalling on application layer. In order to avoid additional stalling caused by the test users' Internet connection, the videos had to be downloaded completely to the browser cache before playing. This enables us to specify fixed unique stalling patterns and initial delays which are evaluated by several users. In particular, we varied the number of stalling events as well as the length of a single stalling event, the length of initial delays, but also the video characteristics in the tests.

During the initial download of the videos, a personal data questionnaire was completed by the participant which also includes consistency questions from above. This personal data allows analyzing the impact on user level. Data was collected concerning the background of the user by integrating demographic questions, e.g. about age or profession. Further, the users were asked to additionally rate whether they liked the content. To get insights into the user's expectations and habits in the context of YouTube, we additionally estimated the user's access speed by measuring the time for downloading the video contents and the users were asked about the frequency of Internet and YouTube usage.

The user then sequentially viewed three different YouTube video clips with a predefined stalling pattern. After the streaming of the video, the user was asked to give his current personal satisfaction rating during the video streaming. In addition, we included gold standard, consistency, content and mixed questions to identify reliable subjective ratings. The workers were not aware of these checks and were not informed about the results of their reliability evaluation. Users had to rate the impact of stalling during video playback on a 5-point **absolute category rating (ACR) scale** with the following values: (1) bad; (2) poor; (3) fair; (4) good; (5) excellent. To be more precise, we asked the users the following question "Did you experience these stops as annoying?" with following answer choices: (5) "imperceptible", (4) "perceptible", (3) "slightly annoying", (2) "annoying", (1) "very annoying".

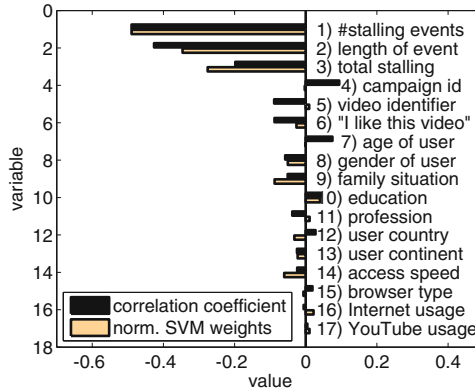
The Microworkers.com platform was used for the crowdsourcing-based QoE tests at the University of Würzburg, since Microworkers allows conducting online user surveys like our YouTube QoE tests. Microworkers supports workers internationally in a controlled fashion, resulting in realistic user diversity well-suited for QoE assessment. The Microworkers platform had about 350 thousand registered users world-wide in the mid of 2012 (see [22] for a detailed analysis of the platform and its users). In total, we conducted seven crowdsourcing campaigns focusing on stalling only and three campaigns focusing on initial delays, respectively. The payment for the crowdsourcing test users was less than 200 €. As described in detail in [21], unreliable test user ratings were identified and the users were filtered out. Throughout the stalling campaigns, 1,349 users from 61 countries participated in the YouTube stalling test and rated the quality of 4,047 video transmissions suffering from stalling. For the initial delay tests, 686 users rated 4,116 videos.

User Studies in Laboratory Environment. In order to validate the crowdsourcing test results and the filtering of unreliable user ratings, similar experiments were carried out in the 'i:lab' laboratory at FTW in Vienna. The user ratings from the crowdsourcing and the lab experiments lead to similar quantitative results and conclusions, see [23, 24]. For the sake of completeness, we shortly describe the lab experiments focusing on initial delays which results are depicted in Figure 8. All other numerical results concerning user ratings stem from crowdsourcing tests as described above.

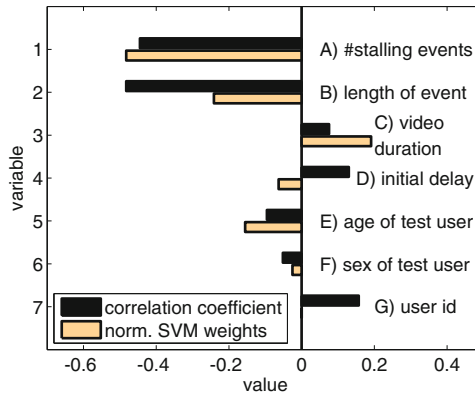
The lab experiment on initial delays contains 41 conditions. The experiment had a total duration of 1.5 h, with an active QoE testing part of about 1 h. The test duration also included a 5 min break in-between the tests and a comprehensive briefing phase at the beginning of the test. Additionally, subjects had to fill out questionnaires about their background, technical experience as well as the current condition, fatigue and cognitive load. After the active testing part, each test was finalized with a debriefing interview and a demographic questionnaire. The QoE testing part consisted of short video clips with a duration of 30 s and 60 s. We used clips out of five different content classes: action trailer, music, animation, documentation and news. After each clip participants were asked to rate the perceived overall quality, including video quality and loading performance, using a 5-point ACR scale on an electronic questionnaire. In total, we collected data from 36 Austrian adults (19 male, 17 female) aged between 20 and 72 years (mean 39.16, median 36.5), recruited by public announcements.

3.2 Key Influence Factors

When it comes to predicting QoE of YouTube, an essential step is determining those key factors that have the strongest influence on the actual experience. Therefore, we analyze correlation coefficients as well as support vector machine (SVM) weights [21]. The Spearman rank-order correlation coefficient between the subjective user rating and the above mentioned variables is computed. In addition, SVMs are utilized to obtain a model for classification: Every variable gets a weight from the model indicating the importance of the variable. However, SVMs are acting on two-class problems only. Therefore, the categories 1 to 3 of the ACR scale to the "bad quality" class and the categories 4 to 5 to the "good quality" class.



(a) Stalling parameters, video characteristics, demographics [21]



(b) Stalling parameters, video duration, initial delays [24]

Fig. 7. Identification of key influence factors on YouTube QoE

Figure 7a shows the results from the key influence analysis. On the x-axis, the different influence factors are considered, while the y-axis depicts the correlation coefficient as well as the SVM weights which are normalized to the largest correlation coefficient for the sake of readability. We can clearly observe from both measures that *the stalling parameters dominate and are the key influence factors*. It is interesting to note that the user ratings are statistically independent from the video parameters (such as resolution, video motion, type of content, etc.), the usage pattern of the user, as well as its access speed. In particular, we used different video contents, but got no differences on the user ratings. A possible explanation may be that a video user does not care about the video quality (i.e. which encoding scheme is used, what is the video bit rate, etc.). If the video

stalls, the video experience of the user is disturbed – independent of the actual video characteristics. Thus, for a YouTube QoE model, those video characteristics can be neglected – in contrast to the actual stalling pattern. However from a networking point of view, higher video bitrates lead to more stalling events if there is a bottleneck in the network. Hence, the video bit rate may be considered for QoE monitoring (see Section 4.2) to estimate the number of stalling events (which is the relevant for the QoE model).

However, the videos considered in the experiments [21] had a fixed length of 30 s and no initial delays for buffering the video contents were considered. Therefore, a second row of experiments was conducted [24] in which the following parameters were varied: number of stalling events, duration of a single stalling event, video duration, initial delay. Hence in addition to the first row of subjective studies, the video duration and initial delays were considered as influence factors. To check again the influence of user demographics on QoE, the age, sex, and user id were also considered in the analysis.

The results in Figure 7b reveal again that *the number of stalling events together with the stalling length are clearly dominating the user perceived quality*, while the user demographics have no influence. Furthermore, the impact of initial delays is statistically not significant. We take a closer look at initial delays that may be accepted by the user for filling up the video buffers to avoid stalling. In case of bad network conditions, providers need to select a trade-off between these two impairment types, i.e. stalling or initial delays, which allows QoE management for YouTube video streaming clouds [25]. Therefore, we ask the question whether initial delays are less harmful to QoE than stalling events for YouTube. However, the results in Figure 8 show that no statistical differences are observed for video clips of 30 s and 60 s regarding the impact of initial delays on the QoE. QoE is thereby quantified in terms of **Mean Opinion Scores (MOS)** which is the average value of user ratings on the ACR scale for a given test condition, i.e. a certain initial delay in that case.

3.3 A Model for QoE Monitoring

The identification of key influence factors has shown that YouTube QoE is mainly determined by stalling frequency and stalling length. To quantify YouTube QoE and derive an appropriate model for QoE monitoring, we first provide mapping functions from stalling parameters to MOS values. Then, we provide a simple model for YouTube QoE monitoring under certain assumptions. Finally, we highlight the limitations of the model.

QoE Mapping Functions. As fundamental relationship between the stalling parameters and QoE, we utilize the IQX hypothesis [26] which relates QoE and QoS impairments x with an exponential function $f(x) = \alpha e^{-\beta x} + \gamma$. In [21], concrete mapping functions for the MOS values depending on these two stalling parameters, i.e. number N of stalling events and length L of a single stalling event, were derived. To be more precise, YouTube videos of 30 s length were considered in the bottleneck scenario leading to period stalling events. In order to determine the parameters α, β, γ of the exponential function, nonlinear regression was applied by minimizing the least-squared errors between the exponential function and the MOS of the user ratings. This way we obtain the best parameters for the mapping functions with respect to goodness-of-fit.

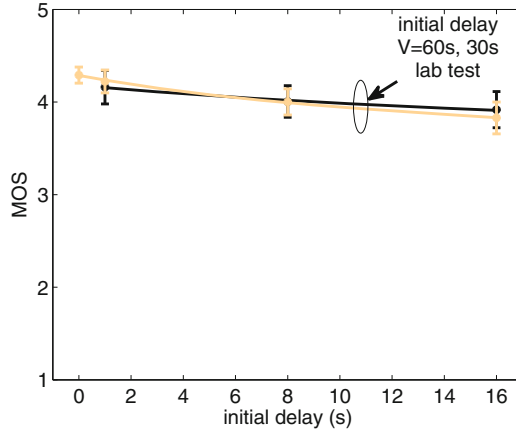


Fig. 8. Initial delays have almost no influence on MOS for videos of duration 60 s and 30 s – compared to influence of stalling length [24]

In this work, however, the aim is to derive a model for monitoring YouTube QoE. Therefore, we reduce the degree of freedom of the mapping function and fix the parameters α and γ . If we consider as QoS impairment x either the number of stalling events or the stalling duration, we observe the following upper and lower limits for the QoE $f(x)$, i.e. $\lim_{x \rightarrow 0} f(x) = \alpha + \gamma$ and $\lim_{x \rightarrow \infty} f(x) = \gamma$, respectively. In case of no stalling, i.e. $x = 0$, the video perception is not disturbed and the user perceives no stalling. As we asked the user “Did you experience these stops as annoying?”, the maximum MOS value is obtained, i.e. $\alpha + \gamma = 5$. In case of strong impairments, however, i.e. $x \rightarrow \infty$, a well-known rating scale effect in subjective studies occurs. Some users tend to not completely utilize the entire scale, i.e. avoiding ratings at the edges leading to minimum MOS values around 1.5 [27]. Hence, we assume $\alpha = 3.5, \gamma = 1.5$ and derive the unknown parameter β from the subjective user tests. The obtained mapping functions as well as the coefficient of determination R^2 as goodness-of-fit measure are given in Table 3. In particular, the mapping function $f_L(N)$ returns the MOS value for a number N of stalling events which have a fixed length L . It can be seen that R^2 is close to one which means a very good match between the mapping function and the MOS values from the subjective studies.

Figure 9 depicts the MOS values for one, two, three and four seconds stalling length for varying number of stalling events together with exponential fitting curves (as discussed in [26]). The x-axis denotes the number of stalling events, whereas the y-axis denotes the MOS rating. The results show that users tend to be highly dissatisfied with two or more stalling events per clip. However, for the case of a stalling length of one second, the user ratings are substantially better for same number of stalling events. Nonetheless, users are likely to be dissatisfied in case of four or more stalling events, independent of stalling duration. As outlined in the previous chapter of this book “From Packets to People: Quality of Experience as a New Measurement Challenge”, most of the users accept a quality above 3 on the ACR scale, i.e. a fair quality.

Table 3. Parameters of mapping functions (see Figure 9) of stalling parameters to MOS together with coefficient of determination R^2 as goodness-of-fit measure

event length L	mapping function depending on number N of stalling events	R^2
1 s	$f_1(N) = 1.50 \cdot e^{-0.35 \cdot N} + 3.50$	0.941
2 s	$f_2(N) = 1.50 \cdot e^{-0.49 \cdot N} + 3.50$	0.931
3 s	$f_3(N) = 1.50 \cdot e^{-0.58 \cdot N} + 3.50$	0.965
4 s	$f_4(N) = 1.50 \cdot e^{-0.81 \cdot N} + 3.50$	0.979

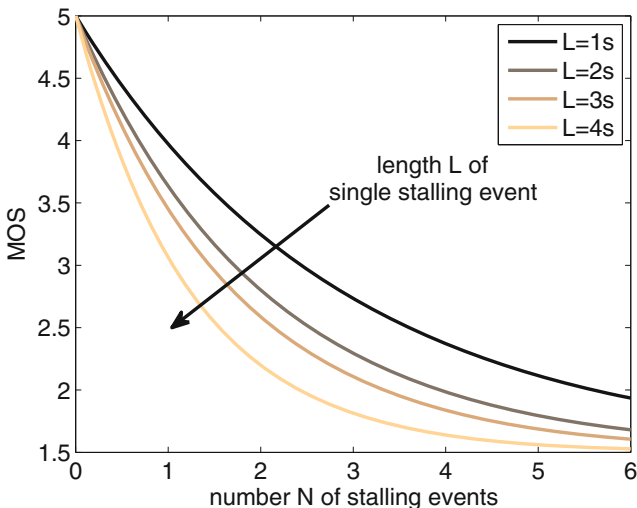


Fig. 9. Mapping functions of stalling parameters to MOS [21]. Video duration is fixed at 30 s. No initial delay is introduced. Parameters are given in Table 3.

It has to be noted that it is not possible to characterize the stalling pattern by a simple total stalling duration $T = L \cdot N$ only, as the curves for $f_L(N)$ depending on the total stalling duration $T = L \cdot N$ differ significantly [20]. Therefore, *stalling frequency and stalling length have to be considered individually in the QoE model.*

Simple Model for QoE Monitoring. Going beyond the pure mapping functions, we develop next an appropriate QoE model for monitoring. The intention of the monitoring is to provide means for QoE management [20] for ISPs or the video streaming service provider. Hence, the model has to consider an arbitrary number N of stalling events and stalling event length L , while the subjective user studies and the provided mapping functions $f_L(N)$ in the previous section only consider a finite number of settings, i.e. $L \in \{1, 2, 3, 4\}$ s. As a result of the regression analysis in the previous section, the parameter β_L of the exponential mapping function $f_L(N) = 3.5^{\beta_L N} + 1.5$ is obtained as given in Table 3.

Figure 10 shows the obtained parameter β_L depending on the length L of a single stalling event. As four settings for L were tested in the subjective user studies, the figure contains four different dots (L, β_L) together with the actual 95 % confidence interval for the nonlinear least squares parameter estimates β_L . Except for the obtained parameter value β_3 for $L = 3$ s, all other values β_L lie on a straight line. Furthermore, the confidence interval of the parameter β_3 overlaps the line. Therefore, we assume in the following a linear relationship between β_L and the event length L which can be easily found as $\beta(L) = 0.15L + 0.19$.

As simple QoE model $f(L, N)$, we therefore combine our findings, i.e. $f_L(N)$ and $\beta(L)$, into a single equation taking the number of stalling events N and the stalling length L as input

$$f(L, N) = 3.50e^{-(0.15L+0.19) \cdot N} + 1.50 \quad \text{for } L \in \mathbb{R}^+, N \in \mathbb{N}. \quad (\text{QoE})$$

Figure 11 illustrates the obtained model for YouTube QoE monitoring as surface plot. On the x-axis the number N of stalling events is depicted, on the y-axis the stalling event length L , while the actual MOS value $f(L, N)$ according to Eq.(QoE) is plotted on the z-axis. The figure clearly reveals that the number of stalling events determines mainly the QoE. Only for very short stalling events in the order of 1 s, two stalling events are still accepted by the user with a MOS value around 3. For longer stalling durations, only single stalling events are accepted.

Other Influence Factors and Limitations of the Model. The scope of this section is to clearly summarize the assumptions and limitations of the provided QoE model as long as the implications for YouTube QoE monitoring. First, we analyze the impact of stalling on YouTube videos of two different durations of $V = 30$ s and $V = 60$ s, respectively. We consider now a single stalling event only ($N = 1$) and vary the stalling duration L from 0 s to 8 s. Figure 12 shows the exponential fitting functions $g_V(L) = 3.5^{\beta_L \cdot L} + 1.5$ of the user ratings in the subjective tests. We see that the curves g_{30} and g_{60} are deviating significantly from each other. Thus, the MOS for the same stalling duration shows significant differences for different video durations: a video with a stalling event of length 8 s is rated 3.30 and 2.51 for a video duration of 60 s and 30 s respectively. Therefore, the video duration must also be taken into account in a proper QoE model. However, it has to be noted that the video duration only plays a role if there are only a very few stalling events (compared to the video duration). Otherwise, the actual number of stalling events dominates the user perceived quality. Nevertheless, Figure 12 depicts the curve for the QoE model in Eq.(QoE) and reveals some limitations of the provided QoE model for monitoring. For longer video durations, the MOS is higher compared to shorter clips with same stalling patterns. Hence, the provided QoE model (obtained for short video clips of 30 s) underestimates the actual QoE. Therefore, QoE monitoring based on this model will give some lower bounds which is desired for QoE management and the avoidance of any QoE degradation of the video streaming service.

Additional assumptions and limitations of the provided QoE model are as follows. In the subjective tests only typical YouTube video formats were considered, however, more

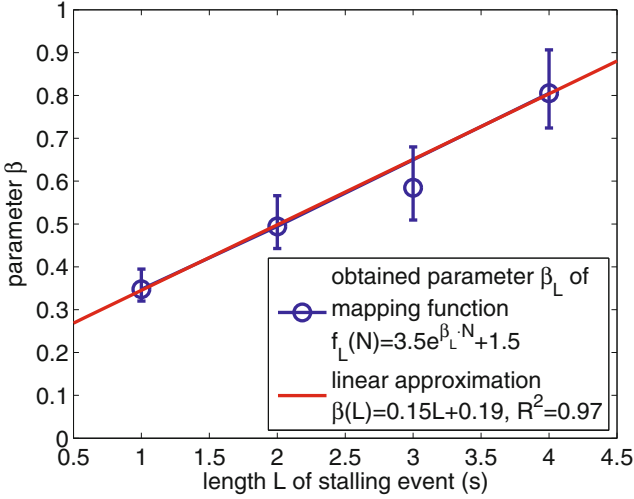


Fig. 10. Parameter β_L of obtained mapping function for given length L of single stalling event. A linear approximation yields a high goodness-of-fit R^2 close to 1

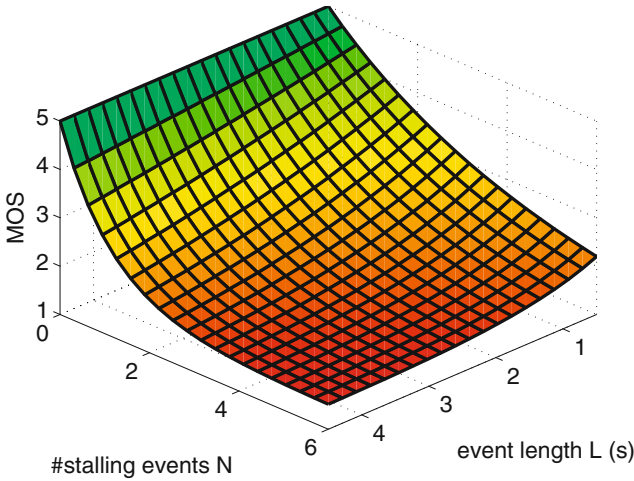


Fig. 11. Simple QoE model maps a number N of stalling events of average length L to a MOS value, $f(L, N) = 3.50 \cdot e^{-(0.15 \cdot L + 0.19) \cdot N} + 1.50$

“extreme” scenarios e.g. very small resolution vs. HD resolution are a subject of future work. Furthermore, the applied stalling pattern considers a bottleneck in the network or at the server side with a constant data rate. This leads to a periodic stalling pattern in which the duration of a single stall event has a fixed duration [21]. Arbitrary stalling patterns due to networks with varying bottleneck capacities are not investigated so far due to the lack

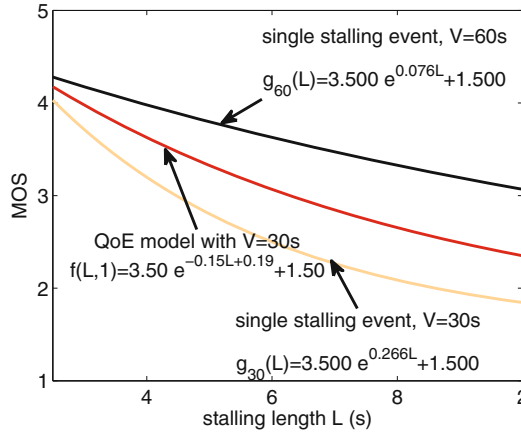


Fig. 12. Influence of stalling length on MOS for video duration of 60 s and 30 s, respectively [24]. A single stalling event is considered without any initial delays.

of related arbitrary stalling models and corresponding subjective user studies. Hence, the impact of different traffic patterns on YouTube QoE is also a matter of future research. As mentioned above, a general relationship between the ratio of the stalling duration and the video duration onto YouTube QoE still is also of interest. Further assumptions of the model are the exponential relationship according to the IQX hypothesis with fixed parameters α and γ due to limits of the MOS values. In addition, we assume a linear relationship between the parameter $\beta(L)$ and the length of a stalling event L .

In summary, *the concrete MOS values depend on the video duration and the actual stalling pattern*. From a QoE management perspective, stalling must be avoided to keep YouTube users satisfied. Even very short stalling events of a few seconds already decrease user perceived quality significantly. However, initial delays are tolerated up to a reasonable level. Hence, YouTube QoE monitoring should pro-actively detect imminent stalling, so that QoE control mechanisms are triggered timely in advance – which is possible with the provided QoE model in the previous section.

4 Monitoring YouTube QoS and QoE

This section provides an overview of different QoE monitoring approaches that have been investigated for YouTube video streaming. As already discussed in the previous section, YouTube QoE is primarily determined by stalling effects on the application layer, as opposed to UDP-based video streaming common for traditional IPTV services. Consequently, the central challenge for QoE monitoring is the *robust detection of stalling events* as they occur during playback. With the number of stalling events and the average length of a single stalling event as input parameters, the QoE model in Eq.(QoE) can be applied to quantify the user perceived quality.

In this context, we distinguish between two categories of monitoring approaches, which differ in terms of measurement point and layer at which the information is captured. **Client**-level monitoring approaches require modifications of the player application

or the installation of additional monitoring clients, but can provide exact results since performance (i.e. stalling patterns) is directly measured at the end user's terminal at the application layer. **Network**-level approaches aim to detect impairments solely from measurements within the network, typically performed at ISP level. We discuss several network-layer monitoring approaches suitable for estimating YouTube QoE, as well as their accuracy (compared to application-level monitoring) and implementation aspects.

4.1 Client-Level Monitoring

As far as client-level monitoring is concerned, most of the related work has been conducted in the context of QoE optimization and traffic management. Staehle et al. [28] present a client-side software tool to monitor YouTube traffic at the application level, by estimating buffer levels to predict stalling events. This approach has been successfully applied to the application-aware self-optimization of wireless mesh networks in [29] in such a way that in case of likely stalling additional resources are provided by the network. In a similar way, the "Forwarding on Gates" (FoG) approach has been used in [30] to develop a novel dynamic network stack based on functional blocks for optimizing the QoE of YouTube playback. The underlying monitoring approach termed **YoMo** is discussed in the following.

C1. YoMo— An Application-level Comfort Monitoring Tool. The YouTube monitoring tool YoMo [28] measures the video player buffer status directly at the end user site on application layer. This allows predicting an imminent stalling taking into account user interactions as pausing the video or jumping within the video, but requires an instance of YoMo running at the user device. Additional challenges for monitoring at the end user level address privacy concerns of users as well as trust and integrity issues due to cheating users to obtain better performance by fraud. However, monitoring at the end-user gives the best view on user perceived quality and YoMo itself is a lightweight Java tool cooperating with a Firefox plugin. The YoMo tool performs several tasks [28]:

- Detect a YouTube flow and forward this information to the mesh advisor which is able to trigger adequate resource management tools in order to avoid a QoE degradation.
- Analyze the packets of the YouTube flow to calculate the video buffer status β .
- Constantly monitor β and raise an alarm if this falls below a critical threshold.

YoMo monitors the client's incoming traffic and identifies a YouTube video flow by detecting the FLV signature. The data of each YouTube video flow is continuously parsed in order to retrieve the information embedded in the FLV tags. In particular, each FLV tag includes the time when the frame is to be played out, allowing to derive the currently available playtime T preloaded in the video buffer.

YoMo estimates the **buffer level** β by computing the difference between playtime T and current **video position** t . However, while the **playtime** T can be derived as the time stamp of the last completely downloaded FLV tag, the current video position t cannot be obtained from the FLV tags, but from the YouTube player only which can be accessed by the YouTube API with scripting languages. Therefore, YoMo uses a simple

Firefox extension which retrieves t from the YouTube player and sends it to the YoMo software. In [28] further technical details are described extensively. Moreover, the paper shows that in the case of a sudden connection interruption, YoMo is able to predict the time of the video stalling event with an accuracy of about 0.1 s. YoMo and the Firefox plugin may be downloaded from the G-Lab website¹.

C2. Pytomo – Analyzing Playback Quality of YouTube Videos. In the context of traffic characterization and measurement of playback performance, another client-level monitoring has been developed by Juluri *et al.* [31]: Pytomo.

The purpose of this Python-based tool is to measure the download and analyze the playback performance of YouTube videos. To this end, it emulates user by downloading a given YouTube video and then selects a number of random related links for further downloads. In this respect Pytomo is not a monitoring tool running in the background like YoMo but rather a *crawler* that actively downloads content in order to perform its measurements. Pytomo’s crawling process can be summarized as follows:

- an initial set of YouTube videos is chosen (by default the most popular videos of the week);
- for each selected video, the cache-URL of the video server hosting the clip file is obtained;
- the (possibly different) IP addresses of the video server are obtained by querying three different DNS servers;
- the ping statistics are collected for each resolved IP address of the video servers;
- from each resolved IP address the first 30 seconds of the video is downloaded at the default video format (640x390)²;
- the crawl continues with the next video.

For each video download, Pytomo collects the following statistics: ping statistics, video information, download statistics, playback statistics such as initial buffer, number of stalling events, total buffering duration, buffer duration at the end of the download (see [31] for a detailed description). Similar to the findings of section Section 3, the authors recommend the number of stalling events and total buffering duration as main indicators for playback QoE.

Like YoMo, Pytomo estimates the playout buffer level by comparing video playtime with the video position. To derive the stalling patterns, this information feed into a model of the YouTube player. While Pytomo relies on the same application-level quality predication approach, it can be considered complementary to YoMo: as a crawler, it actively measures YouTube QoE with high repeatability, albeit at its current state does not predict end user QoE. Furthermore, Pytomo allows to study the impact of the DNS resolver used on playback quality, which as shown in [31], can have a considerable impact. Pytomo is a GPL software and can be freely downloaded from <http://code.google.com/p/pytomo>.

¹ <http://www.german-lab.de/go/yomo>

² In order to perform a video download, Pytomo first resolves the IP address of the content server and then uses this IP address to perform the analysis. This ensures that the analysis and video download are being performed on the same server.

4.2 Network-Level Monitoring

While being highly accurate, client-level approaches are not applicable in the case of an ISP interested in monitoring YouTube QoE inside its network. The main reason is that the installation of additional software on the client side as well as the migration to a non-standard network stack or topology are not practical options.

In the case of YouTube, the main challenge is the accurate approximation of the stalling events that take place at the application layer using network packet traces only, which is non-trivial. One example of work in this field is [32], where the authors present an approach for measuring YouTube stalling events from packet traces. The paper presents some first interesting results, but is too limited in terms of number of analyzed videos to draw conclusions on the accuracy of the approach. Moreover, no QoE models or estimations are derived from this analysis. In contrast, other works focus on monitoring YouTube stalling events and estimating the corresponding QoE levels, relying exclusively on passive monitoring of TCP flows. Previously in [33], we have presented and compared three different passive monitoring approaches, referred to as 'M1', 'M2', and 'M3'. The first approach M1 is based on the download time of the whole video; M2 relies on measuring the end-to-end throughput of the connection; finally, M3 approximates the actual video buffer status. All three approaches allow to estimate the stalling pattern without relying on application-level or client side measurements, however with different levels of accuracy and complexity. In general, the stalling pattern can be described by the **total stalling time** T , the **number of stalling events** N , and the duration or **length of a stalling event** L . In case of a bottleneck with constant capacity B , regular stalling events are observed as measured in [19]. Assuming a known distribution for L , we can formulate the first and most simple monitoring approach:

M1. Download Time vs. Video Duration. In this approach, the monitoring system measures the total stalling time T as the difference between the total video download time Y and the video duration D , i.e. $T = \max(Y - D, 0)$. With a given average stalling length L , the number N of stalling events can be roughly approximated as $N = T/L$ (cf. [33]). Note that the download time of the video contents can be easily extracted from packet-level traces.

A first problem with M1 arises, when trying to obtain the overall duration of the video D . There are several possibilities in practice. First, this information is available from the YouTube website and can be requested directly via the YouTube API. Therefore, the monitoring system has to extract the YouTube video identifier from the HTTP request containing the url and the video id. An alternative option is to extract the information from the video header. YouTube uses for example the FLV container file format, from which meta data like video duration, frame rate and key frames are specified. In that case, the monitoring system has to parse the network packets and needs to understand the container format. Hence, both options require some extra effort for the system to get the video duration. However, the major drawback of M1 is that it uses the *total* duration of the video as input. Indeed, if the user does not watch the entire video and thus aborts downloading before the end, which is very frequent in practice, this monitoring approach cannot be applied. For this reason, a more complex approach is required for passive probing scenarios.

M2. Network Throughput vs. Video Encoding Rate. The second approach is based on the stalling frequency approximation in [21]. Based on a considerable body of measurement data, the authors show that the frequency F of stalling events can be well approximated with an exponential function

$$F(x) = -1.09e^{-1.18x} + 0.36 \tag{1}$$

with x being the normalized video demand $x = V/B$ defined as ratio of video encoding rate V and download throughput B . In that case, the bottleneck capacity B has to be estimated, which can be easily done from passive monitoring packet traces and throughput measurements [34]. Furthermore, the video bitrate V has to be extracted from the packets by parsing the meta data available at the container file format. From these two values, the normalized video demand $x = V/B$ is computed. Finally, the number of stalling events can be approximated by $N = \min(D, Y) F(x)$, where Y represents in this case the current download time of the video, and not the total video download time as in M1. Similar to M1, the video duration D has to be extracted from the packet traces.

The computational effort for M1 and M2 is comparable, but M2 can also be applied to cases where the user does not download and watch the whole video content. However, the major problem with M1 and M2 is accuracy. Both approaches estimate either the total stalling time T and/or the number of stalling events N , assuming that the distribution of the stalling length L is known. For example, L can be approximated by a t location-scale distribution [19]. Although the length of a single stalling event lies between 2 s and 5 s with high probability, this approach leads to inaccurate results and thus QoE estimations with considerable errors.

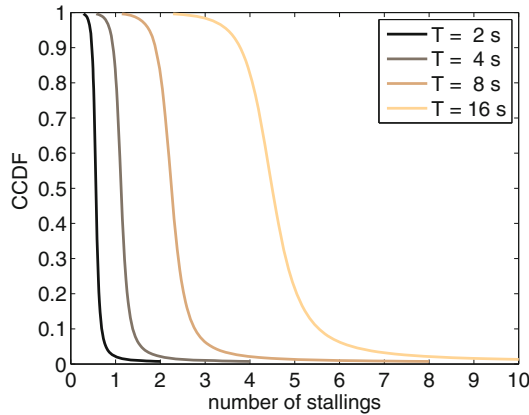


Fig. 13. Monitoring approaches M1 and M2 can only estimate the number of stalling events with a certain probability [33]

Figure 13 shows the complementary cumulative distribution function of the number of stalling events $N = F^{-1}$ estimated for given total stalling times T , which are varied from 2 s to 16 s. The estimation is obtained by using the t -location-scale approximation,

and the approximation applied in M1, i.e., $N = T/L$. This estimation of N clearly exhibits a large variance, particularly for longer total stalling time values. For example, for a total stalling time of 8 s, the number of stalling events lies between 2 and 4 with high probability. However, this range already has a strong impact on the actual QoE, ultimately deciding between good and bad quality: as shown in [21], the number of stalling events is crucial for the end-user experience. Thus, the QoE differences for N and $N + 1$ stalling events may be dramatic. For example, for $N = 1$ the difference is about 0.7 – 0.8 MOS in a 5-point MOS-scale. Consequently, in practice an ISP can use both approaches only for upper or lower bound estimations of QoE.

For some ISP, it may be sufficient to determine whether stalling occurs or not and how the user reacts in response. Therefore, we define the QoS metric **reception ratio** ρ as ratio between download throughput B and video encoding rate V , i.e.

$$\text{reception ratio } \rho = \frac{B}{V}.$$

Although the reception ratio cannot be directly related to QoE, it is a good indicator if there are problems in the network. We demonstrate this by relating the reception ratio to the user behavior based on our work in [11], which is one of the first attempts to quantify the impact of playback quality on the viewing behavior.

A download throughput lower than encoding rate should result in interrupted playback: thus we define the **reception quality** as QoE indicator in the following way,

- if reception ratio > 1 , we consider the video has **good reception quality**;
- otherwise we consider the video has **poor reception quality**.

This metric may first seem quite crude, so we have used active measurements from [15] to evaluate its accuracy. The dataset used consists of eight packet traces of one hour collected between 2008 and 2011 (see [11] for details). We use two standard metrics usually used in pattern recognition and information retrieval, namely **precision** and **recall**. They are based on the concepts of

- *True Positive TP*: reception ratio > 1 and the video had no stall;
- *False Positive FP*: reception ratio > 1 but the video had at least one stall;
- *True Negative TN*: reception ratio < 1 and the video had at least one stall;
- *False Negative FN*: reception ratio < 1 but the video had no stall.

Out of these notions, we build these evaluation metrics:

- **recall** = $TP / (TP + FN)$: this corresponds to the fraction of uninterrupted videos correctly evaluated;
- **precision** = $TP / (TP + FP)$: this corresponds to the ratio of uninterrupted videos in the videos with reception ratio > 1 .

For the data set of December 2011 we obtain 91.8% of recall and 88.5% of precision. While the overall performance of this indicator is surprisingly good, it suffers from the

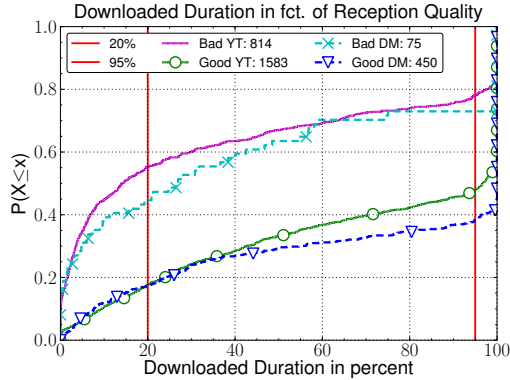


Fig. 14. Fraction of video downloaded as function of video reception quality for YouTube (YT) and Dailymotion (DM) (from [11])

following two limitations [19]. First, if the video duration is very short, sufficient data is downloaded before the video playout starts and no stalling occurs. Second, stalling is caused by the variability of the video bit rate, which may happen even when the network capacity is larger than the video bit rate.

Furthermore, an ISP is interested in the relation between the user perception and the user behavior. Therefore, we analyze in the following the fraction of the video downloaded (as user behavior indicator) depending on the reception quality (as QoE indicator).

In Figure 14, we plot the CDF of the *fraction of the video downloaded*. We distinguish between videos with good reception quality and those with poor reception quality. We see that under good playback quality, less than 50% YouTube transfers are aborted before the end, whereas with poor playback quality, as much as 80% of the transfers are aborted before the end. We have also included another popular video streaming site, namely DailyMotion. The results are quite similar for both video streaming sites, which justify the claim that reception quality influences the viewing behavior.

In Figure 15, we distinguish each video according to its duration and the fact that it has been completely downloaded (at least 90% of video downloaded):

- short videos (≤ 3 minutes);
- long videos (≥ 3 minutes) and completely downloaded;
- long videos (≥ 3 minutes) and not completely downloaded;

We plot the downloaded duration vs. the content duration for YouTube videos: in Figure 15a for videos with good reception quality, and in Figure 15b for videos with poor reception quality. We observe that in case of good reception quality, 34% of the videos have a downloaded duration of 3 minutes or more, while in case of poor reception quality their share drops to only 15%. This confirms that the reception quality influences the behavior of the user, and this influence is more pronounced for long videos.

M3. YiN- YouTube in Networks Based on Playout Buffer Approximation. The third passive monitoring approach M3, also called “YiN” in [33], detects YouTube video flows like for client-level monitoring (see Section 4.1). It extracts video information from network packet data. In particular, the size and time stamps of (audio and video) frames are retrieved by means of deep packet inspection. Together with the YouTube video player parameters in particular the playing threshold Θ_1 and the stalling threshold Θ_0 (see Figure 6), the playout video buffer status is reconstructed on behalf of network data only at high accuracy. As soon as the YouTube video buffer exceeds Θ_1 , the player starts the video playback. If the buffer underruns Θ_0 , the video stalls. The player parameters are determined in [33] based on the application-layer measurements in [19]. However, it has to be noted that there may small deviations of these values from video to video in practice, since the player takes into account the actual structure of the video codec for optimized video playout. Consequently, such small errors may propagate and lead to inaccuracies in practice.

The basic idea of YiN is to compare the playback times of video frames and the time stamps of received packets. We define the frame time τ_i as follows. After receiving the i -th acknowledgment on TCP layer at time t_i , a total amount of $\nu = \sum_{j=1}^i \nu_j$ bytes has been downloaded. Together with the size of each video frame and the video frame rate – typically around 25 frames/s –, the frame time τ_i corresponds to the downloaded video ‘duration’ so far. Then, we define the play time ρ_i and the stalling time σ_i to be the user experienced video play time and stalling time after the i -th TCP acknowledgment. The actual amount of buffered video time is indicated by β_i . The boolean stalling variable ψ_i indicates whether the video is currently playing ($\psi_i = 0$) or stalling ($\psi_i = 1$).

On behalf of these measures the stalling pattern over time, i.e. over the TCP acknowledgments, can be computed as follows [33].

$$\psi_i = \psi_{i-1} \wedge \beta_{i-1} < \Theta_0 \vee \neg\psi_{i-1} \wedge \beta_{i-1} < \Theta_1 \quad (2)$$

$$\sigma_i = \sigma_{i-1} + \begin{cases} t_i - t_{i-1}, & \text{if } \psi_i \\ 0, & \text{if } \neg\psi_i \end{cases} \quad (3)$$

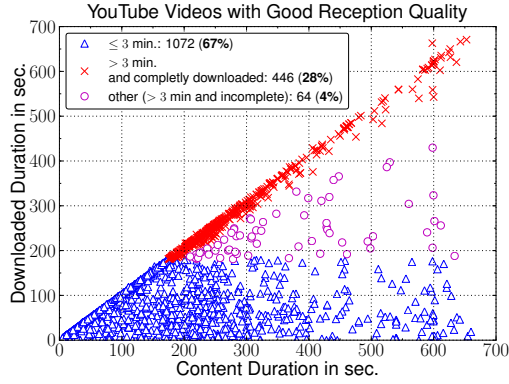
$$\rho_i = \rho_{i-1} + \begin{cases} 0, & \text{if } \psi_i \\ t_i - t_{i-1}, & \text{if } \neg\psi_i \end{cases} \quad (4)$$

$$\beta_i = \tau_i - \rho_i \quad (5)$$

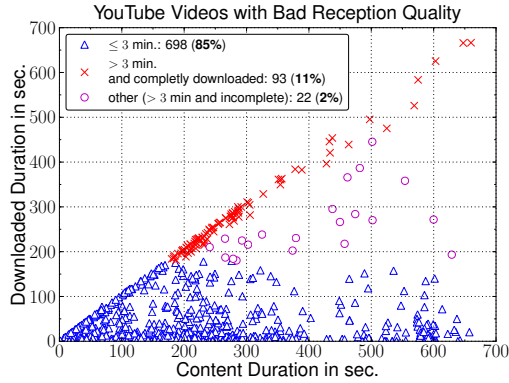
The actual video buffer can then be approximated by the difference between the frame time τ_i and the actual play time ρ_i . The iterative computation of the different variables is initialized in the following way, since YouTube first starts playing until the threshold Θ_1 is exceeded to fill the video buffer.

$$\sigma_0 = 0, \quad \rho_0 = 0, \quad \psi_0 = 1. \quad (6)$$

QoS and QoE-Level Evaluation of Network Monitoring Approaches. As already outlined above (cf. Figure 13), the monitoring approaches M1 and M2 can only estimate the number of stalling events with a certain probability. Hence, their accuracy is not sufficient for proper QoE monitoring. In this context, the concept of reception ratio as



(a) Good reception quality videos



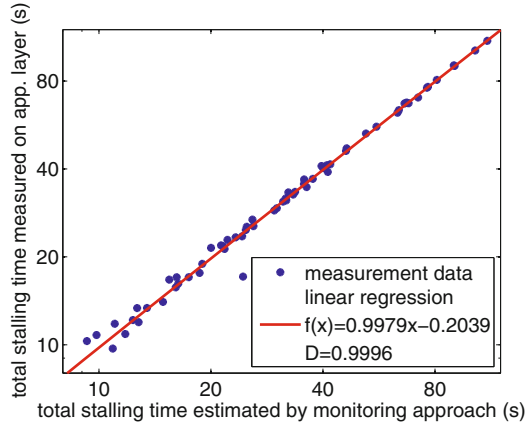
(b) Bad reception quality videos

Fig. 15. Fraction of video downloaded as function of video length (from [11])

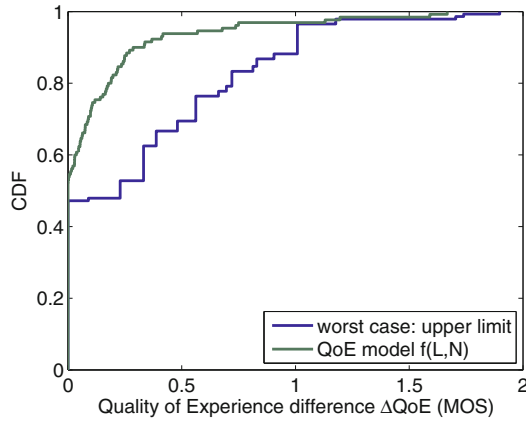
ratio between download throughput and video encoding rate was introduced to indicate whether stalling occurs or not.

To evaluate the accuracy of the YiN monitoring approach, which aims to retrieve the stalling pattern, the estimated video buffer was compared with the actual video buffer measured at application layer, which serves as ground truth. The measurements took place from June 2011 to August 2011 in a laboratory at FTW in Vienna (see [33] for further details). Furthermore, in a second step the stalling patterns were mapped to QoE according to the YouTube QoE model (Section 3.3) and the difference between 'measured' and 'estimated' QoE based on the reconstructed stalling patterns were compared.

In this respect, the results in [33] show that YiN is the most accurate approach that predicts that stalling pattern almost exactly with a coefficient of correlation between measured and estimated values of about 0.9998. Figure 16a illustrates this very strong correlation between the total stalling time estimated from packet traces and the total stalling time as measured on the application-layer.



(a) QoS: Measured vs. estimated total stalling times



(b) QoE: Difference ΔQoE between measurements on application layer and estimation

Fig. 16. Network-level monitoring YiN (from [33]): Comparison of application-layer measurements and approximations based on network-level information

Nevertheless, the remaining measurement error can lead to strong QoE differences due to the end user’s non-linear perception of stalling that is also reflected in the QoE model discussed in Section 3.3. We compare our results considering two different evaluation scenarios: (i) the QoE model evaluation using the real stalling patterns, and (ii) a worst case evaluation, which provides an upper bound to the QoE estimation difference. For the worst case evaluation, short stalling events of 1 s length are considered, which sum up to the total stalling time. This is a worst case scenario because it leads to a higher number of stalling events than actually observed. Again, the difference ΔQoE between ‘measured’ and ‘estimated’ QoE based on the reconstructed stalling patterns are compared.

The cumulative distribution functions of the ΔQoE values obtained in both scenarios are depicted in Figure 16b. The QoE difference w.r.t. the QoE model is almost zero for about 60% of the analyzed videos. As worst case upper bound, two thirds of the analyzed videos show a QoE difference below 0.5. However, differences can be as large as one step on the MOS scale, as observed for 10% of the videos. Thus, the YiN monitoring approach may estimate good quality (MOS 4), while the users actually only experience a fair quality (MOS 3). The main reason for these inaccuracies is – as described above – error propagation; since according to end-user quality perception and the underlying mapping from stalling QoS to YouTube QoE are highly non-linear, a relatively small measurement error can result in aforementioned MOS differences. For example, when the number of stalling events is very low, one stalling more or less already makes a huge difference in QoE. As a consequence, one has to take these error margins into account and set alarm thresholds accordingly [20].

Nonetheless, the above results demonstrate that an accurate reconstruction of stalling events from network-level measurement data is possible, and that YouTube QoE monitoring at ISP-level is feasible. However, only the most accurate and unfortunately the most complex YiN approach can be actually used for QoE monitoring purposes, since (i) stalling frequency and stalling duration both need to be measured, and (ii) the non-linearity of human perception demands for high QoS measurement accuracy, particularly in those cases where stalling frequency is low.

5 Conclusion

In this chapter, we have presented the YouTube delivery infrastructure and we have investigated YouTube video streaming in terms of QoE impact of Internet delivery as well as resulting QoE monitoring aspects. To this end, we discussed the key mechanisms used by YouTube: Cache selection plays an important role in YouTube and we showed that surprisingly, cache server selection is highly ISP-specific and that geographical proximity is not the primary criterion, for reasons that need further investigation. In addition, DNS level redirections for load-balancing purposes occur quite frequently and can considerably increase the initial startup delay of the playback (in the order of seconds). However, the results from subjective user studies showed that initial delays up to about ten seconds have no severe impact on QoE. Hence, QoE models and QoE monitoring approaches may neglect those initial delays.

From a QoE management perspective, the smooth playback of the video rather than visual image quality is the key challenge, since YouTube uses HTTP via TCP to deliver the video. We saw that, more than any other impairment, stalling events (i.e. playback interruptions) have a dramatic impact on the QoE and should be avoided at any price. The monitoring of the stalling frequency and duration is the prerequisite for proper QoE monitoring. In this context, the throughput-based reception ratio plays a key role as QoE-relevant metric for predicting buffer under-runs. Concerning QoE monitoring, we compared several network-level and client-level approaches with regard to their accuracy to detect stalling pattern. As expected, this is more difficult for network-level approaches, which have to reconstruct client-level stalling patterns from network traffic information only. However, our evaluation of the YiN algorithm shows that this is feasible, albeit at the cost of increased demand for computational performance.

As far as future work is concerned, the QoE management for video streaming to smartphones remains an open issue. The mobile environment will lead to different traffic and stalling patterns that need to be evaluated from a QoS and QoE monitoring perspective accordingly.

Acknowledgements. The results presented in this chapter summarize a successful research period starting 2010. The different facets discussed in this chapter (subjective test design, crowdsourcing, QoE model, monitoring approaches, CDN structure) were supported by different research projects. In particular, the research has been supported by COST TMA Action IC0703 through two short term scientific missions by T. Hoßfeld (“Time Dynamic Modeling of Quality of Experience (QoE) for Web Traffic” and “QoE-based Bandwidth Dimensioning for ISPs to Support Online Video Streaming”); COST QUALINET Action IC1003 through the STSM “Modeling YouTube QoE based on Crowdsourcing and Laboratory User Studies” by T. Hoßfeld; by the European FP7 Network of Excellence “Euro-NF” through the Specific Joint Research Project “PRUNO”; and the project G-Lab, funded by the German Ministry of Educations and Research (Förderkennezeichen 01 BK 0800, G-Lab). In addition, this work has been supported within the projects ACE 2.0 and U-0 at the Telecommunications Research Center Vienna (FTW) and has been funded by the Austrian Government and the City of Vienna within the competence center program COMET. This work was funded by Deutsche Forschungsgemeinschaft (DFG) under grants HO 4770/1-1 and TR257/31-1. The authors alone are responsible for the content of the paper.

The authors would like to give special thanks to Alexander Platzer and Pedro Casas for the fruitful discussions on YouTube monitoring approaches; Michael Seufert and Matthias Hirth for implementing, running and improving the crowdsourcing experiments for QoE tests; and Sebastian Egger for all discussions on YouTube QoE and carrying out the laboratory tests. Further, we would like to thank the responsible editor Maja Matijasevic and the anonymous reviewers for their valuable comments and suggestions to improve this chapter.

References

1. Cisco: Cisco visual networking index: Forecast and methodology, 2011–2016 (May 2012)
2. Plissonneau, L., Vu-Brugier, G.: Mobile data traffic analysis: How do you prefer watching videos? In: Proc. of 22th International Teletraffic Congress (ITC'22), Amsterdam, Netherlands (September 2010)
3. Maier, G., Feldmann, A., Paxson, V., Allman, M.: On dominant characteristics of residential broadband internet traffic. In: Proc. of 9th ACM SIGCOMM Internet Measurement Conference (IMC 2009), Chicago, Illinois, USA (November 2009)
4. Kontothannis, L.: Content Delivery Consideration for Web Video. In: Keynote at ACM Multimedia Systems 2012 (MMSys 2012), Chapel Hill, North Carolina, USA (2012)
5. Brodersen, A., Scellato, S., Wattenhofer, M.: YouTube Around the World: Geographic Popularity of Videos. In: Proc. of 21th International World Wide Web Conference (WWW 2012), Lyon, France (April 2012)
6. Ghobadi, M., Cheng, Y., Jain, A., Matthis, M.: Trickle: Rate Limiting Youtube Video Streaming. In: Proc. of 2012 USENIX Annual Technical Conference, Boston, MA, USA (June 2012)

7. Adhikari, V.K., Jain, S., Chen, Y., Zhang, Z.L.: Vivisecting YouTube: An Active Measurement Study. In: Proc. of IEEE INFOCOM 2012 Mini-Conference, Orlando, Florida, USA (March 2012)
8. Adhikari, V.K., Jain, S., Zhang, Z.L.: YouTube Traffic Dynamics and Its Interplay with a Tier-1 ISP: An ISP Perspective. In: Proc. of 10th ACM SIGCOMM Internet Measurement Conference (IMC 2010), Melbourne, Australia (November 2010)
9. Adhikari, V.K., Jain, S., Zhang, Z.L.: Where Do You “Tube”? Uncovering YouTube Server Selection Strategy. In: Proc. of International Conference on Computer Communications and Networks (ICCCN 2011), Maui, Hawaii, USA (August 2011)
10. Finamore, A., Mellia, M., Munafo, M., Torres, R., Rao, S.: YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience. In: Proc. of 2011 ACM SIGCOMM Internet Measurement Conference (IMC 2011), Berlin, Germany (November 2011)
11. Plissonneau, L., Biersack, E.: A Longitudinal View of HTTP Video Streaming Performance. In: Proc. of ACM Multimedia Systems 2012 (MMSys 2012), Chapel Hill, North Carolina, USA (February 2012)
12. Torres, R., Finamore, A., Kim, J.R., Mellia, M., Munafo, M.M., Rao, S.: Dissecting video server selection strategies in the youtube cdn. In: Proc. of 31st International Conference on Distributed Computing Systems (ICDCS 2011), Minneapolis, Minnesota, USA (June 2011)
13. Zhou, J., Li, Y., Adhikari, V.K., Zhang, Z.L.: Counting YouTube videos via random prefix sampling. In: Proc. of 2011 ACM SIGCOMM Internet Measurement Conference (IMC 2011), Berlin, Germany (November 2011)
14. Rao, A., Legout, A., Lim, Y., Towsley, D., Barakat, C., Dabbous, W.: Network characteristics of video streaming traffic. In: Proc. of 7th International Conference on emerging Networking Experiments and Technologies (CoNEXT 2011), Tokyo, Japan (December 2011)
15. Plissonneau, L., Biersack, E., Juluri, P.: Analyzing the Impact of YouTube Delivery Policies on the User Experience. In: Proc. of 24th International Teletraffic Congress (ITC'24), Krakow, Poland (September 2012)
16. Alcock, S., Nelson, R.: Application flow control in YouTube video streams. *ACM SIGCOMM Computer Communication Review* 41(2) (April 2011)
17. Hoßfeld, T., Schatz, R., Biedermann, S., Platzer, A., Egger, S., Fiedler, M.: The Memory Effect and Its Implications on Web QoE Modeling. In: Proc. of 23rd International Teletraffic Congress (ITC'23), San Francisco, USA (September 2011)
18. Schatz, R., Egger, S., Hoßfeld, T.: Understanding Ungeduld - Quality of Experience Assessment and Modeling for Internet Applications. In: Proc. of 11th Würzburg Workshop on IP: Joint ITG and Euro-NF Workshop Visions of Future Generation Networks (EuroView 2011), Würzburg, Germany (August 2011)
19. Hoßfeld, T., Zinner, T., Schatz, R., Seufert, M., Tran-Gia, P.: Transport Protocol Influences on YouTube QoE. Technical Report 482, University of Würzburg (July 2011)
20. Hoßfeld, T., Liers, F., Schatz, R., Staehle, B., Staehle, D., Volkert, T., Wamsler, F.: Quality of Experience Management for YouTube: Clouds, FoG and the AquareYoum. *PIK - Praxis der Informationsverarbeitung und -Kommunikation (PIK)* 35(3) (August 2012)
21. Hoßfeld, T., Schatz, R., Seufert, M., Hirth, M., Zinner, T., Tran-Gia, P.: Quantification of YouTube QoE via Crowdsourcing. In: Proc. of IEEE International Workshop on Multimedia Quality of Experience - Modeling, Evaluation, and Directions (MQoE 2011), Dana Point, CA, USA (December 2011)
22. Hirth, M., Hoßfeld, T., Tran-Gia, P.: Anatomy of a Crowdsourcing Platform - Using the Example of Microworkers.com. In: Proc. of Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2011), Seoul, Korea (June 2011)

23. Hoßfeld, T.: Towards YouTube QoE via Crowdsourcing. Online Lecture within COST Qualinet (November 2011),
http://www3.informatik.uni-wuerzburg.de/papers/tutorials_16.zip
24. Hoßfeld, T., Schatz, R., Egger, S., Fiedler, M., Masuch, K., Lorentzen, C.: Initial Delay vs. Interruptions: Between the Devil and the Deep Blue Sea. In: Proc. of 4th International Workshop on Quality of Multimedia Experience (QoMEX 2012), Yarra Valley, Australia (July 2012)
25. Hoßfeld, T., Schatz, R., Varela, M., Timmerer, C.: Challenges of QoE Management for Cloud Applications. *IEEE Communications Magazine* 50(4) (April 2012)
26. Fiedler, M., Hoßfeld, T., Tran-Gia, P.: A generic quantitative relationship between quality of experience and quality of service. *IEEE Network - Special issue on Improving Quality of Experience for Network Services* 24(2) (March 2010)
27. Tominaga, T., Hayashi, T., Okamoto, J., Takahashi, A.: Performance comparisons of subjective quality assessment methods for mobile video. In: Proc. of 2nd International Workshop on Quality of Multimedia Experience (QoMEX 2010), Trondheim, Norway (July 2010)
28. Staehle, B., Hirth, M., Pries, R., Wamser, F., Staehle, D.: YoMo: A YouTube Application Comfort Monitoring Tool. In: Proc. of EuroITV Workshop QoE for Multimedia Content Sharing (QoEMCS 2010), Tampere, Finland (June 2010)
29. Staehle, B., Hirth, M., Pries, R., Wamser, F., Staehle, D.: Aquarema in Action: Improving the YouTube QoE in Wireless Mesh Networks. In: Proc. of Baltic Congress on Future Internet Communications (BCFIC 2011), Riga, Latvia (February 2011)
30. Hoßfeld, T., Liers, F., Volkert, T., Schatz, R.: FoG and Clouds: Optimizing QoE for YouTube. In: Proc. of KuVS 5thGI/ITG KuVS Fachgespräch NG Service Delivery Platforms, Munich, Germany (October 2011)
31. Juluri, P., Plissonneau, L., Medhi, D.: Pytomo: a tool for analyzing playback quality of YouTube videos. In: Proc. of 23rd International Teletraffic Congress (ITC'23), San Francisco, USA (September 2011)
32. Rugel, S., Knoll, T., Eckert, M., Bauschert, T.: A network-based method for measurement of internet video streaming quality. In: Proc. of 1st European Teletraffic Seminar, Poznan, Poland (February 2011)
33. Schatz, R., Hoßfeld, T., Casas, P.: Passive YouTube QoE Monitoring for ISPs. In: Proc. of Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2012), Palermo, Italy (July 2012)
34. Lai, K., Baker, M.: Nettimer: A Tool for Measuring Bottleneck Link Bandwidth. In: Proc. of USENIX Symposium on Internet Technologies and Systems (USITS 2001), San Francisco, California, USA (March 2001)

Quality Evaluation in Peer-to-Peer IPTV Services

Mu Mu¹, William Knowles¹, Panagiotis Georgopoulos¹, Steven Simpson¹,
Eduardo Cerqueira², Nicholas Race¹, Andreas Mauthe¹, and David Hutchison¹

¹ Lancaster University, Lancaster, LA1 4WA, United Kingdom
{m.mu,w.knowles,p.georgopoulos,s.simpson,n.race,a.mauthe,
d.hutchison}@lancaster.ac.uk

² University of Pará, Belém, Brazil
cerqueira@ufpa.br

Abstract. Modern IPTV services are comprised of multiple comprehensive service elements in the entire content delivery chain to maximise the efficiency in networking. Audio-visual content may experience various types of impairments during content ingest, processing, distribution and reception. While some impairments do not cause noticeable distortions to the delivered content, many others such as the network transmission loss can be highly detrimental to the user experience in content consumption. In order to optimise service quality and to provide a benchmarking platform to evaluate the designs for future audio-visual content distribution system, a quality evaluation framework is essential. We introduce such an evaluation framework to assess video service with respect of user perception, while supporting service diagnosis to identify root-causes of any detected quality degradation. Compared with existing QoE frameworks, our solution offers an advanced but practical design for the real-time analysis of IPTV services in multiple service layers.

Keywords: Quality of Experience, IPTV, Peer-to-Peer, Living Lab.

1 Introduction

Assuring the quality of user experience in an IPTV service is challenging, especially when complex distribution mechanisms such as Peer-to-Peer (P2P) are exploited. In practice, different types of impairments exist in service entities including source content ingest, video coding, packet networks, P2P overlay, and the end system in a P2P-based IPTV system. While some impairments can be corrected and concealed, others such as the transmission loss lead to annoying distortions to the delivered video content, which are highly detrimental to the user experience. For instance, a surge of network usage in a home network may introduce network impairments that limit the throughput of an IPTV set-top box, starve the decoder buffer and eventually cause break-ups to the TV programmes. When distortions are repeatedly experienced, disappointed customers must explain the time and phenomenon of distortions to customer service while

user devices do not provide any features for retrospective quality analysis. Therefore, it is impossible for service providers to efficiently pin-point the root-cause of service interruptions.

In order to fulfil user expectation of service quality and to provide a benchmarking platform that evaluates designs for future audio-visual content distribution system, content and service providers, a quality evaluation service is required. This evaluation service must provide real-time assessment of video service with respect of user perception while supporting service diagnosis to effectively identify root-causes of any detected quality degradation. Most existing QoE frameworks are designed for laboratory evaluations and therefore not suitable in practice. This article introduces a quality evaluation service as well as its implementation in a Living Lab IPTV environment with real users. Section 2 briefly introduces the IPTV service. Section 3 summarises the challenges and requirements of quality assessment in an IPTV platform. The framework of our evaluation service is given in Section 4 followed by the details of the measurement and analysis modules. An use case is provided in Section 5. Section 6 concludes our work.

2 Living Lab IPTV Services

The role of the Lancaster University Living Lab is to function as a small and open service provider for the purpose of research and real-life evaluation of state-of-the-art technologies. This process encompasses the implementation and operation of technical services, the provision of user devices, and the subsequent measurement and evaluation procedures.

At the beginning of the IPTV service chain is the process of content ingest. The headend receives and de-multiplexes live DVB-T (terrestrial) and DVB-S (satellite) channel signals over-the-air including UK channels as well as various continental European channels in multiple languages.

Following the content ingest, two groups of virtual machines prepare the content for live and on-demand services respectively. The live service takes the source video streams and injects them into the peer-to-peer (P2P) platform. This produces the P2P version of the stream, along with the associated torrent file for playback. The Living Lab uses the NextShare P2P engine which is a BitTorrent-based P2P distribution engine with a number of new features that are specifically designed for streaming of audio-visual content. In order to address the incompatibility of BitTorrent's tit-for-tat mechanism and sequential downloading (e.g., as when streaming media), the NextShare introduces a novel give-to-get mechanism, which "discourages free riding by letting peers favour uploading to peers who have been proven to be good uploaders" [8].

The IPTV video-on-demand (VoD) service records programmes from the corresponding channels and stores them on a SAN (storage area network) back-end. The VoD service is also delivered using the NextShare P2P distribution platform. A torrent file for each programme is automatically created and announced to the tracker.

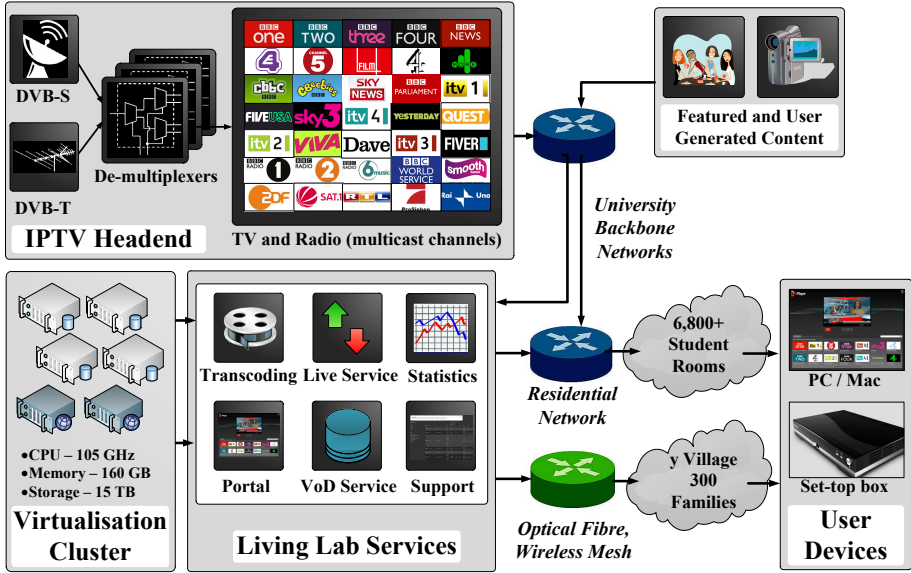


Fig. 1. Living Lab IPTV service infrastructure

On the user side, the Living Lab services can be received on two derivative application platforms NSPC and NSTV which target the PC and consumer electronics (CE) respectively. NSTV provides an integrated and easy to use consumer electronics device showcasing the feasibility and benefits of the NextShare platform. It is based on the STB7200 system-on-chip technology provided by ST Microelectronics. NSPC is a multi-platform component which provides web browser integration to the NextShare utilising Web 2.0 technologies to provide a mixture of dynamic AJAX-enabled HTML pages and XML feeds. The IPTV service is accessible to over 6000 student and staff users on university campus.

3 Challenges, Requirements and Related Work

3.1 Challenges

Commercial IPTV services have strict QoE requirements to guarantee a high content delivery standard as specified by service level agreement (SLA) recognised between service providers and content consumers. In [4] the quality requirements for IPTV services are specified. It is concluded that video streams are highly vulnerable to information loss and that the QoE impact is in turn correlated to a number of variables including: type of data loss, codec, loss profile and decoder concealment algorithms. QoE requirements must therefore be defined with regard to discrete quality violation events [12]. In the ITU Focus Group on IPTV, quality target metrics have been extended to consider the actual quality as perceived by end users. User level quality metrics such as “Maximum one visible

artifact per × hours” were defined to evaluate the delivery of IPTV services [1]. The Broadband Forum has also defined a set of user level QoE metrics in [15]. For example, a criterion of “*one impairment event per 12 hours or better*” is defined for HDTV services.

The quality of user experience is ultimately determined by users’ subjective opinions. To collect users’ opinions, subjective experiments are usually carried out using representative test materials and well-specified test plans in dedicated test environments (e.g., [3]). However, conventional subjective experiments are time-consuming, costly and therefore not suitable for an in-service evaluation, especially in commercial IPTV services.

3.2 Requirements

A number of objective quality evaluation models have been designed to conduct quality assessment without human intervention and participation. Objective models analyse the decoded audio-visual content at the receiver and evaluate audio and visual distortions using audio, image and video signal processing tools [19,2]. However, realising such a analysis at end systems in operational services is extremely challenging. The computational complexity of most audio-visual analysis algorithms usually exceeds the capacity of any consumer device. Furthermore, most objective quality models are not designed to identify the causes and network locations of quality degradation.

Network QoS models are widely used to evaluate the impact of network impairments to services. Some advanced QoS models have been recently developed by integrating application-level metrics [20]. Discrete analysis of perceptual impact of individual packet loss has also been designed to better capture user-level QoE [12,10]. However, these models are not suitable to evaluate impairments within the P2P overlay layer and within end systems. Furthermore, the compression distortions caused by lossy video coding can not be evaluated directly using network-layer models.

Overall, the level of evaluation given by existing QoE models are not sufficient to fully support quality assessment and service diagnosis for IPTV services. The objective is only achievable by the collaboration of multiple quality models, with each providing a required type of evaluation function. Service diagnosis is enabled by comparing and matching the time-coded evaluation results of relevant quality models. For instance, when an objective distortion measurement model detects a visual artefact in video frames, measurement from both network and end system can be used to efficiently identify the cause of this distortion. If the cause has been the decoder error, specific actions can be taken to examine decoding procedures.

3.3 Related Work

Designing an operational QoE service for a IPTV service has different requirements as the conventional QoE framework or testbed. For instance, EvalVid [7] is a framework with a well-defined tool-set for the evaluation of the quality of video

transmitted over a real or simulated communication network. However, EvalVid is designed for offline laboratory evaluation of network, codec or service design. Many of the QoS metrics and standard video quality metrics like SSIM are not suitable for live evaluations in IPTV networks. A QoE assessment and management framework was designed for real-time multimedia applications [13,9]. The framework employs the most suitable objective analysis functions with respect to specific test conditions. However, it does not support a timed measurement so identifying the root-cause of quality degradation is not possible. Gardikis, G. et al. explored application and network level metrics for cross-layer monitoring in IPTV networks [6], though integration of quality metrics and implementation in practice are not available. The QoM (quality of experience framework for multimedia services) introduced by Rehman Laghari, K. et al. also considers QoE evaluation of multiple layers but the measurement is still carried out manually by Wireshark while the user interface simply adopts the laboratorial user test design [17]. The five-point quality rating interface that is ideal for post service survey would greatly distract end users from viewing the IPTV content. Hence, a solution that could more efficiently collect user response in a discreet manner is required. Alvarez, A. also design a flexible QoE framework to evaluate transmission impairments [5]. The solution is based on the full-reference PSNR metric which is not suitable for real-time quality assessment in a large scale IPTV service.

4 IPTV QoE Framework

4.1 Overview

Figure 2 shows the framework for the evaluation system. Although the quality of user experience is ultimately determined by the condition of the delivered content, heterogeneous measurement tools are strategically placed in IPTV service network to facilitate *early quality assessment* (prior to the user's perception of distortion) and prompt diagnosis. The framework interacts with five key elements of a video distribution system including source content, audio-visual encoder (transcoder), distribution network, end system and end user. Multimodal assessment of service quality is conducted by the collaboration of measurement, analysis and diagnosis modules, which are realised by groups of functional components.

Relevant service metrics from all key elements of the distribution system are extracted and summarised by the measurement module before data analysis and visualisation of the analysis module are initiated. The framework supports both retrospective analysis for measurements captured in a previous time range and real-time analysis for instantaneous reports updated every few seconds. The diagnosis module coordinates analysis results for different measurement functions to enable comprehensive evaluation for service diagnosis. Functional modules and blocks can be selectively activated according to specific test plans and strategies.

Figure 3 illustrates a use case of an evaluation to analyse the influence of network impairments in our P2P-based IPTV services. Using the time-stamp

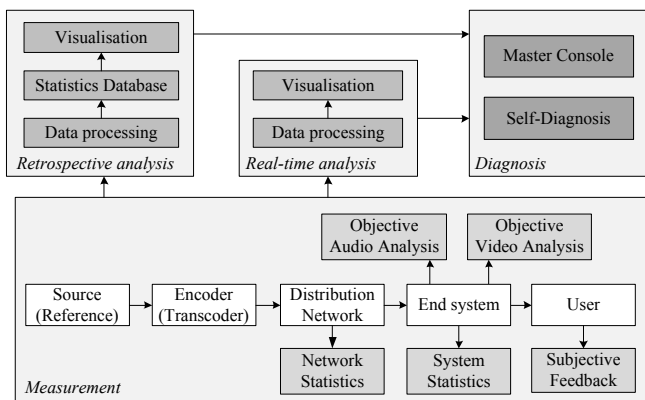


Fig. 2. Framework for multimodal QoE evaluation

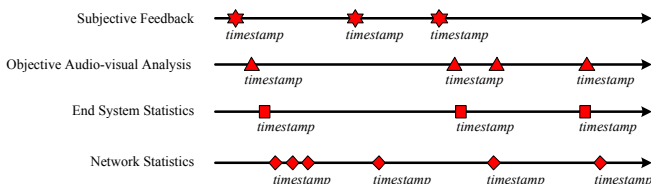


Fig. 3. Multimodal evaluation associated by timestamp information

information associated with all evaluation processes, we are able to correlate pre-defined events that are detected in different layers.

In order to conduct a comprehensive evaluation which studies multiple aspects of video distribution services, a number of functional components have been designed to capture service statistics with respect to transmission, distribution, video codec and human perception. Overall, the measurement module is comprised of *in-service subjective feedback*, *objective audio-video analysis*, *system statistics*, and *network statistics*. The in-service subjective feedback function provides a simple and interactive interface for viewers to report perceived audio and video distortions and to answer optional questionnaires regarding the overall service quality. The objective audio and video analysis function captures decoded audio and video signals from the output of the set-top box and assesses the quality using objective (no reference) quality models. System and network statistics functions report metrics reflect service status regarding video decoding, content distribution and packet-based networks respectively. An earlier architectural design of the multimodal evaluation framework is presented in one of our recent work [11]. The following sections introduce design and recent implementation of each functional module.

4.2 In-Service Subjective Feedback

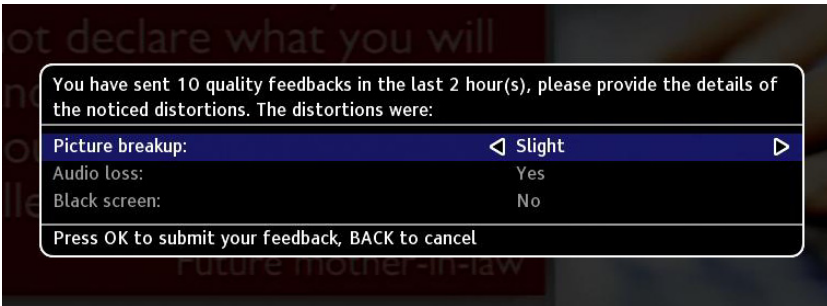
The in-service user feedback function can be embedded in an IPTV set-top box to provide means for end users to efficiently trigger both instantaneous feedbacks and questionnaires regarding the service quality. A user can conveniently report a perceived audio or video distortion such as picture breakup by pressing a dedicated button (e.g., the *blue button*) on the remote controller. A small icon appears on the screen as an acknowledgement of receiving user feedback (Figure 4(a)).



(a) Distortion report



(b) Distortion report close-up



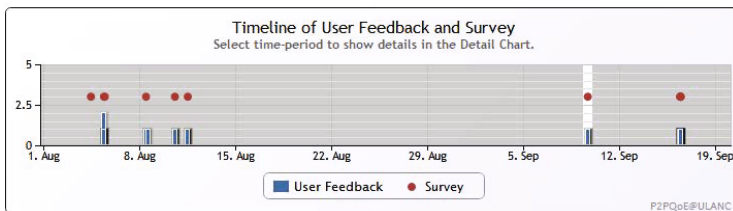
(c) Service questionnaire

Fig. 4. In-service subjective feedback

The overall impact of multiple perceived distortions is also modelled taking into account relevant psychological effects. For instance, a forgiveness effect (a.k.a. memory effect [14]) exists due to the fact that the objection felt by an observer immediately following an impaired video segment is compensated after a long period of unimpaired video [16]. Our system recognises the forgiveness effect using a sliding *attention span*, which defines a certain period of time backwards from the current time. If the number of distortion reports triggered within this attention span (such as 30 minutes) reaches a predefined threshold (such as 5 times), a notification is issued and a questionnaire is initiated for end users to provide details of corresponding service disruption (Figure 4(c)). The questions and options of questionnaire, the trigger number, and the length of attention window are defined using a XML configuration file located on a designated server

MAC	IP	Date/Time	Action	Media Address	Media Type	IsLive	Content Duration	Playback Offset
00E036A1154F	148.88.227.181	2011-06-30 01:30:59	1	http://p2pnext-swarm-one.lancs.ac.uk/live/BBCNEWS24.mpegts.tstream	torrent	1	3600	1679
00E036A1154F	148.88.227.181	2011-06-30 01:30:56	1	http://p2pnext-swarm-one.lancs.ac.uk/live/BBCNEWS24.mpegts.tstream	torrent	1	3600	1676
00E036A1154F	148.88.227.181	2011-06-30 01:30:53	1	http://p2pnext-swarm-one.lancs.ac.uk/live/BBCNEWS24.mpegts.tstream	torrent	1	3600	1673
00E036A1154F	148.88.227.181	2011-06-30 01:30:51	1	http://p2pnext-swarm-one.lancs.ac.uk/live/BBCNEWS24.mpegts.tstream	torrent	1	3600	1671
00E036A1154F	148.88.227.181	2011-06-30 01:30:48	1	http://p2pnext-swarm-one.lancs.ac.uk/live/BBCNEWS24.mpegts.tstream	torrent	1	3600	1668
00E036A1154F	148.88.227.181	2011-06-30 01:30:45	1	http://p2pnext-swarm-one.lancs.ac.uk/live/BBCNEWS24.mpegts.tstream	torrent	1	3600	1665
00E036A1154F	148.88.227.181	2011-06-30 01:30:43	1	http://p2pnext-swarm-one.lancs.ac.uk/live/BBCNEWS24.mpegts.tstream	torrent	1	3600	1663
00E036A1154F	148.88.227.181	2011-06-30 01:30:40	1	http://p2pnext-swarm-one.lancs.ac.uk/live/BBCNEWS24.mpegts.tstream	torrent	1	3600	1660
00E036A1154F	148.88.227.181	2011-06-30 01:30:37	1	http://p2pnext-swarm-one.lancs.ac.uk/live/BBCNEWS24.mpegts.tstream	torrent	1	3600	1657
00E036A1154F	148.88.227.181	2011-06-30 01:30:35	1	http://p2pnext-swarm-one.lancs.ac.uk/live/BBCNEWS24.mpegts.tstream	torrent	1	3600	1655

(a) Table view and screen recreation



(b) Chart view

Fig. 5. Analysis of subjective feedback

and made available to all set-top boxes. This set-up enables dynamic manipulation of subjective feedback interface should there be any customisation required.

Both the instantaneous distortion report and service questionnaire are forwarded to the retrospective analysis module. Details of subjective feedback are extracted and stored in a statistics database. A user interface is also implemented to visualise subjective feedback records. Figure 5(a) shows a recreation of a selected questionnaire recorded in the database and details of all associated distortion reports. An interactive timeline chart application gives a more intuitive representation of user feedback. The blue bars represent the distortion reported whereas the red dots indicate completed questionnaires. Using this tool, investigators can explore responses from customers to facilitate service diagnosis. In practice, a user (such as a child) can repeatedly press the report button even when no error is perceived. The false positive (and false negative) events can be identified when measurement on this module is studied in conjunction with results from other measurement modules such as the system statistics.

4.3 Objective Audio-Visual Analysis

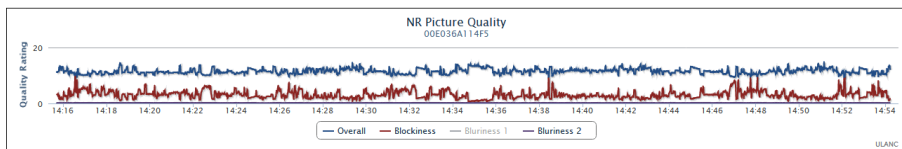
The objective analysis function aims at analysing audio and visual distortions introduced during the life-cycle of content processing and delivery (i.e., coding, network transmission, decoding) by studying the decoded audio-visual signal. The measurement is carried out on the decoded video frames using image signal processing tools and decode audio samples using audio analysis tools. The analysis is carried out using the method of no reference (NR). No reference models identify distortions in the received video content using predefined signatures without the knowledge of the source content. Compared with existing offline QoE

frameworks, the objective analysis employed by our framework does not provide high level mapping such as a QoS to QoE function which would be artificial and unusable for operational services. All analysis are carried out to detection distortions as quality violation events.

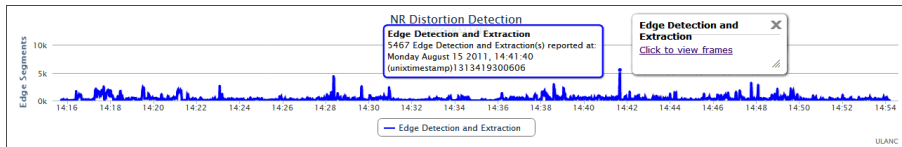
In our IPTV system, digital video signals are transmitted from the set-top box to a display unit using High-Definition Multimedia Interface (HDMI). For signal processing algorithms to process video frames, a video capture device with HDMI input is exploited. In order to conduct objective video analysis and subjective feedback simultaneously, a HDMI-splitter device is employed to duplicate the HDMI output signal from set-top box to form two identical HDMI feeds (one for playback and the other for analysis). The audio sampling for audio quality analysis also follows the same procedure. The audio signal can be sampled either from the HDMI input or from an analog audio input. Because the objective analysis is carried out on the rendered audio-visual signals, the solution is therefore codec independent.

Video Frame Analysis. Objective quality models are commonly designed to recognise and evaluate only certain types of distortions reflecting different quality evaluation strategies. Three quality assessment models are implemented to provide a wide spectrum of video analysis. The *picture quality analysis* model is a realisation of a no-reference perceptual quality assessment algorithm initially designed by Wang [18]. The model measures two main distortions i.e., blurriness and blockiness, caused by lossy video compression on each video frame. Results of distortion measurements are combined using an aggregation function to derive an overall quality rating. Figure 6(a) gives results of a picture quality assessment test. The picture quality analysis model is a valuable tool to evaluate the influence of video encoding/transcoding process to the picture quality. It can be used for benchmarking different transcoding configurations and performance of encoding configurations on various types of content (e.g., action films or news channels).

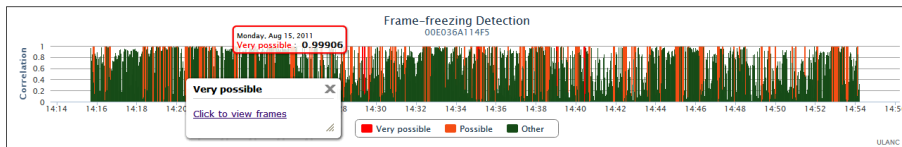
In practice, distortions in video frames can be caused by video compression, transmission impairments or errors in end systems. To better identify the distortions such as severe blockiness and frame-freezing caused by transmission impairments or system errors, edge detection and frame-freezing detection models are also designed and integrated. The edge detection model extracts visual edges appearing at the boundaries of all 16×16 macroblocks that are above a predefined threshold (Figure 7). This process filters out most of the light blockiness (which are usually not visible by end users) caused by video compression and also the edges of objects within videos. Figure 6(b) shows the number of edge units detected on all video frames in a test. Sudden impulses of measurement are considered as potential severe distortions in video. The visualisation chart is also made interactive so that investigators can click on a measurement point to verify the results by visually check corresponding video frames (archived by the analysis function). Figure 8(a) shows the video frame associated with the impulse manifests at 14:41:40 in Figure 7.



(a) Picture quality analysis



(b) Edge detection

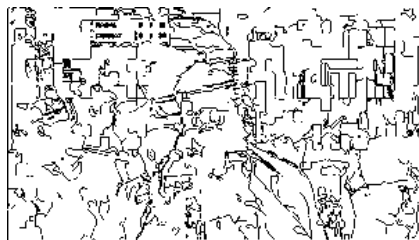


(c) Frame-freezing detection

Fig. 6. Objective video analysis



(a) Severe blockiness distortions



(b) Edge extraction

Fig. 7. Edge detection

The frame-freezing detection function is realised by exploiting the correlation between consecutive video frames captured by the video capture card. The detection function marks the event as *possible* frame-freezing when the correlation reaches 0.98 and *very possible* when the correlation is over 0.99. Results of a frame-freezing detection test is given in Figure 6(c). *very possible*, *possible*, and *others* are marked in distinctive colours. In practice, genuine still scenes exist. Therefore, video frames associated to the events of *very possible* frame-freezing are made available for visual verification. Figure 8(b) shows the interface with which the detected frame-freezing events are verified.

Although the three video frame analysis functions employed by the QoE framework are not specifically heavy in terms of computational complexity, providing live analysis on all frames of a live high definition video is not realistic

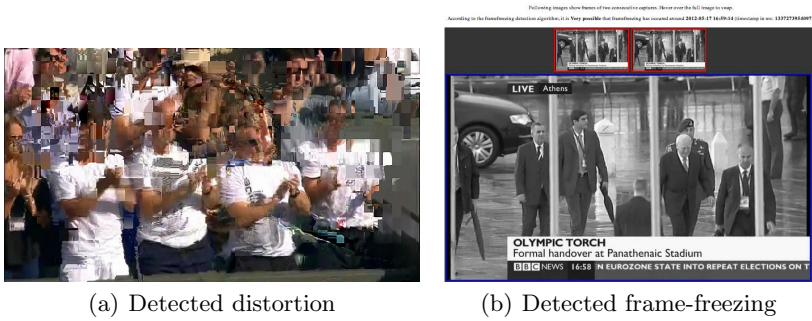


Fig. 8. Visual verification of video analysis

without a high-end measurement device. Therefore, samples are taken on the input video signal using the sampling frequency that is suitable for the hardware adopted for the video analysis. Using the dedicated PC (Intel Core2 Duo Processor E6400 2.13GHz) our Matlab implementation is configured to work on 3 frames of every second of video.

Audio Analysis. One challenge of exploiting video frame analysis for the frame-freezing detection is to differentiate between frame-freezing distortion and genuine still frame as part of the source video. In some TV programmes, still pictures are showing on the screen while voice commentary is given. Without studying the audio track, an objective analysis model would yield high volume of “false positive” distortions. Therefore, the audio analysis is adopted so that a joint measurement can be made between audio and video frame analysis.

The current design of the audio analysis module detects the events when no sound is detected in the video within a certain period of time (e.g., 200 milliseconds). Figure 9 gives an example of a combined analysis of audio and visual

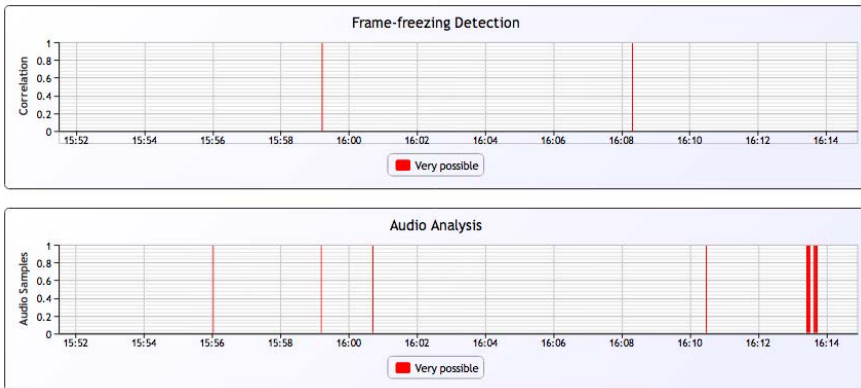


Fig. 9. Combined analysis of audio and visual signals

signals. It is noticed that a few silence events are detected in the video stream. As it is for the video analysis, the audio “freezing” can be either legitimate quiet moments (e.g., a long pause in a conversation) in the video or distortions caused by lose of data. Therefore, objective audio-visual analysis should be conducted by combining the results of audio and video quality measurements.

4.4 System Statistics

Network packets are received and assembled at end systems before P2P pieces are merged for video decoding. A report engine is implemented in our IPTV set-top boxes to accumulate and report system statistics. The system statistics gives details reflecting the software and hardware status of set-top boxes, the handling of incoming packets, the P2P piece turnaround and video decoding process. Examples of system statistics are *box status (playing, standby, off)*, *total number of decoded video frames*, *min, max and average bit-rate*, *buffer overflow and underflow events*, *decoder syntax error* and *piece turnaround time*. Some metrics of system statistics such as the ones that indicate errors reported by decoder are highly correlated to distortions in video frames. Some other metrics, such as the buffer level, provide insights into the root-causes of decoder errors.

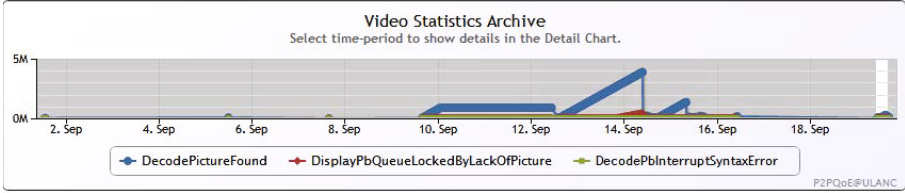
The report of system statistics to a dedicate statistics server is triggered periodically (e.g., every 15 minutes) and also by a number of pre-defined events such as the start and end of media playback. All system statistics are parsed and stored in an archive database for service analysis. A visualisation interface is designed so that investigators can extract statistics based on specific criteria (e.g., device id, IP address, time range, etc.). Figure 10(a) shows a number of system statistics reported by a particular set-top box. In order to better support service diagnosis, a more intuitive timeline chart interface illustrates a few selected metrics of system statistics. Figure 10(b) gives an example of statistics analysis to identify the causes of visual distortions. It is noticed that at around 10:40 on Aug 18, 2011 nearly 20 decode syntax errors were identified. The number of syntax errors increased to 41 at about 11:40, which caused over 200 *display queue lock* events caused by lack of pictures in the queue. This is an indication of severe blockiness or frame-freezing appeared on the user’s screen. The set-top box was restarted afterwards which reset both metrics to zero as captured just before 12:00. A timeline analysis such as the one shown in Figure 10(b) is valuable to establish the correlations between metrics captured at different layers of a IPTV system.

Timely and orderly reception of P2P pieces is essential to the smooth playback of audio-visual content. Requested pieces may arrive out of order or fail to arrive on time, due to a lack of content availability in the P2P network or as a result of network impairments. The *piece request* and *piece receive* time of all P2P pieces are monitored to study the turnaround time (i.e., the time difference between piece request and receive) of every P2P piece, as well as the cases of pieces that fail to arrive. Figure 11 shows the mean P2P piece turnaround time of every second. The figure suggests that there are some minor fluctuations in network traffic from 10:43 onwards.

[This page shows 100 of 271 records in total] First Previous Next Last

uid	mac	ip	timestamp	sversion	powerstate	activity	uptime	total_ram	free_ram	temperature	num_processes	num_keypress	post_error	post_timeout	eventid	cpu_load1	cpu_load2	cpu_load3	main_uid	App
1680	00E36A11509	192.168.1.52	2011-08-30 12:56:25	1.8.0.7709	2	1	9225	169004	60232	53.5	105	71	0	0		7.77	7.16	6.55		0
1681	00E36A11509	192.168.1.52	2011-08-30 12:56:25	1.8.0.7709	2	1	9225	169004	60232	53.5	105	71	0	0		7.77	7.16	6.55		0
1679	00E36A11509	192.168.1.52	2011-08-30 12:49:29	1.8.0.7709	2	1	8749	169004	75260	53.5	115	41	0	0		6.80	6.10	6.90		0
1677	00E36A11509	192.168.1.52	2011-08-30 12:33:29	1.8.0.7709	2	1	7848	169004	76072	53.5	117	57	0	0		6.19	6.24	6.15		0
1668	00E36A11509	192.168.1.52	2011-08-30 12:18:28	1.8.0.7709	2	1	6948	169004	76852	53.0	115	57	0	0		6.60	6.70	6.80		0
1666	00E36A11509	192.168.1.52	2011-08-30 12:03:28	1.8.0.7709	2	1	6047	169004	77976	53.0	115	57	0	0		6.29	6.17	6.16		0
1664	00E36A11509	192.168.1.52	2011-08-30 11:49:28	1.8.0.7709	2	1	5147	169004	78336	52.0	115	57	0	0		6.20	6.20	6.18		0
1662	00E36A11509	192.168.1.52	2011-08-30 11:33:27	1.8.0.7709	2	1	4246	169004	79176	50.5	117	57	0	0		6.30	6.18	6.24		0

(a) Table view



(b) Chart view

Fig. 10. End system statistics archive

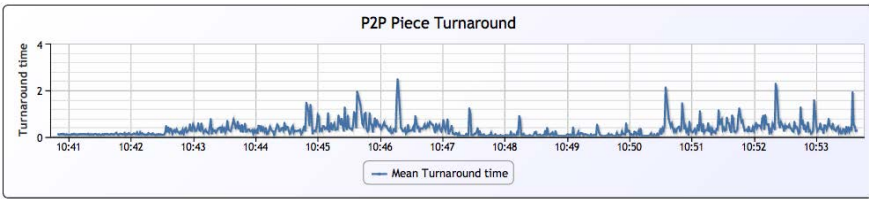


Fig. 11. Mean P2P piece turnaround time

An additional four P2P-layer metrics have been defined, in order to facilitate the detection of events that could potentially affect the received video quality. *Late* and *Drop* are defined according to a prioritised piece download range which shifts with the media playback time. A piece that has arrived beyond the download range is marked as *late*. Pieces that have failed to arrive after a pre-defined threshold are considered lost and registered as *drop*. The *stall* metric provides an estimate of playback stalls by analysing the pieces in the egress queue in the decoder. When the video buffer is effectively empty and incoming data can not keep up with content playback, an *underrun* message is triggered.

4.5 Network Statistics

Network impairments such as packet delay and loss are the main causes of detrimental quality degradation in audio-visual content distribution networks. Meanwhile, events in end systems (e.g., buffer over-flow) and in P2P-layers (e.g., ineffective exchange of pieces) can both cause issues in packet networks (e.g., TCP resets). The objective of network-layer analysis is to inspect and analyse key network-layer metrics in distribution networks to better identify the root-cause of service interruptions as experienced by end users. Due to the nature of P2P-based content distribution, content is assembled from pieces provided by

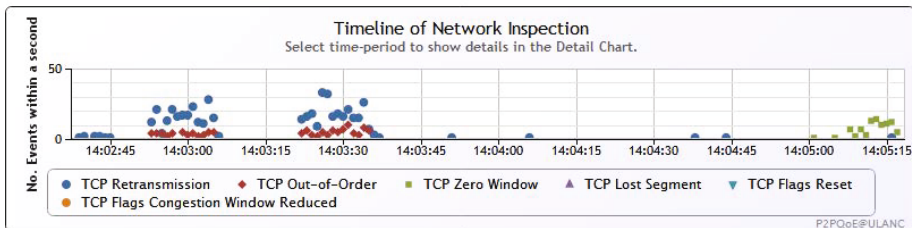


Fig. 12. Network statistics

peers reside in various networks. Therefore, examining network traffic using the traditional network QoS metrics is not practical. In order to measure the TCP-based P2P piece distributions, several TCP transaction capture and analysis tools are implemented at network interface (e.g., an access aggregation point) adjacent to the set-top box. Figure 12 shows a number of network statistics captured, including *TCP retransmission*, *TCP out of order*, *TCP zero window*, *TCP lost segment*, *TCP reset* and *TCP congestion window reduced*. All metrics are defined as the number of events detected within a second.

In order to test system performance under different network conditions, various types of controlled transmission impairments can also be emulated using either traffic control with *Netem* or radio signal management of wireless mesh boxes within our wireless P2P-based distribution services.

Table 1 summaries the update frequency and implementation of all functional modules as they are currently realised in the Living Lab IPTV service.

5 Use Case

Figure 13 shows an ongoing evaluation session which captures statistics in all service layers. A NSTV (i.e. NextShareTV) set-top box is connected to campus network via a gateway (two wireless mesh network devices). Network statistics are captured at the gateway using packet inspection and transmission analysis tools. P2P pieces are assembled and decoded by NSTV box where P2P statistics and System statistics are acquired and posted to statistics server. Digital video signals from NSTV box are displayed on a 40-inch LCD Full-HD TV unit and also fed to a video capture card on an evaluation workstation where a Matlab program conducts objective quality assessment using image acquisition, audio acquisition and four different analysis algorithms. An investigator uses a remote controller to submit instantaneous feedback and answer questionnaires. The test results are analysed using a master QoE analysis console.

While table and chart view of individual measurement is used for analysing a specific service metric (such as buffer level of set-top box), the master console is designed to identify the root-cause of service interruptions. The console integrates multiple visualisation charts and synchronises all relevant charts when a time range is specified. Figure 14 shows how master console is used to identify

Table 1. Current update frequency and implementation of framework modules

Module	Update frequency	Hardware requirements
Subjective Feedback	User press the dedicated button.	Integrated in set-top box.
Audio-visual Analysis	Three samples per second for all three types of analysis.	A dedicated measurement PC with audio and video capture capabilities. Integration in set-top box is possible.
System Statistics	Every 15 minutes or triggered by predefined events.	Integrated set-top box function or browser plug-in.
Network Statistics	Data processing on TCP dump is carried out every 20 minutes.	A dedicated measurement PC for both packet inspection and analysis.

**Fig. 13.** An ongoing in-service quality evaluation

the causes of a number of subjective feedback received around 11:34. The system and frame-freezing detection statistics both report severe errors (“Display queue locked by lack of picture” and “Very possible frame-freezing”) that resonate with users’ responses. This is also verified by visual check (as shown in Figure 8) and survey recreation of user feedbacks (as shown in Figure 5(a)).

Tracing back on the timeline of network statistics from the points of subjective responses, a large number of *TCP retransmission* and *TCP out of order* are recognised around 11:32 and 11:33 (Figure 14). For this specific evaluation test, the cause of frame-freezing distortions as perceived by users is believed to be the result of packet loss and jitter arising from the distribution network.

The use case demonstrates the necessity and effectiveness of a multi-modal evaluation framework for the assessment of a complex audio-visual service like the P2P-based IPTV services of Lancaster Living Lab. Using combinations of measurement tools and metrics, comprehensive analysis for service diagnosis and other research activities have been made possible. As one of the first QoE framework that exploits timed and real-time analysis of IPTV service quality in multiple service layers, there are many lessons learned from the design as well

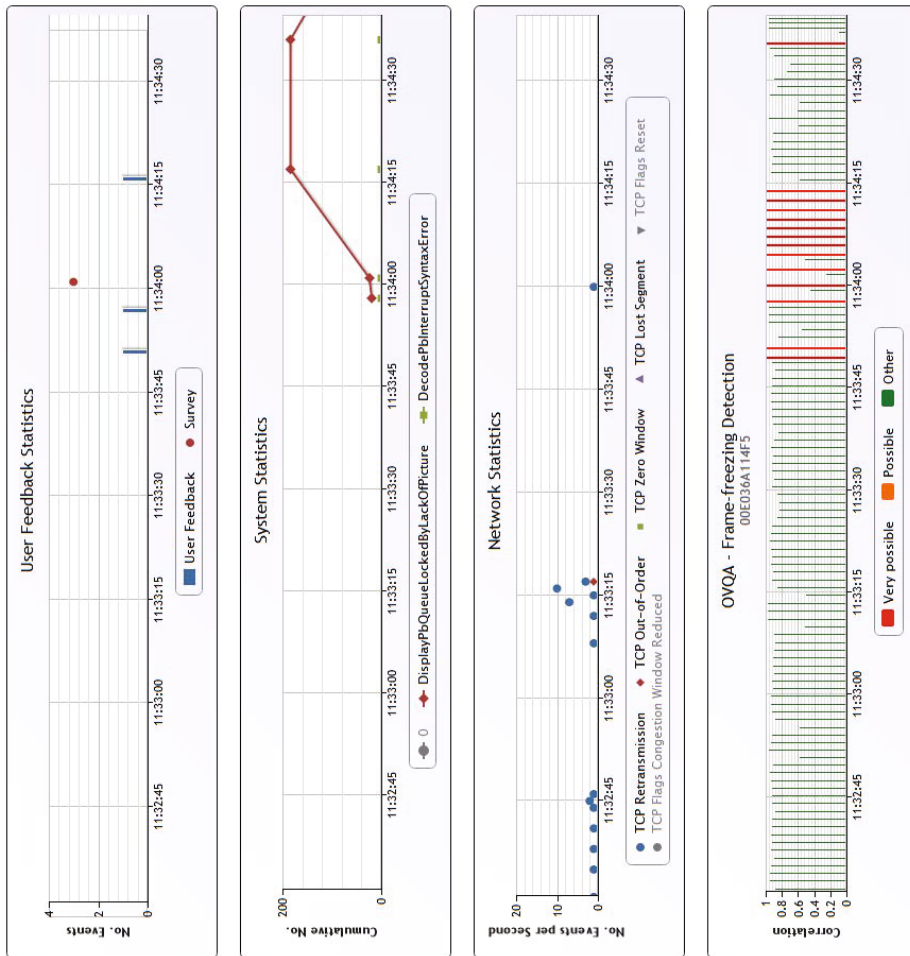


Fig. 14. Master console showing monitoring results (excerpt)

as the deployment phase. The statistics service requires a level of management and maintenance which would lead to extra service cost in practice. However, the statistics can be extremely valuable for quality diagnosis and service level agreement referencing between service providers. The subjective instantaneous feedback design is also well received by our test participants. However, users usually expect any issue to be corrected shortly after it is reported. This brings more requirements to the service support. The no-reference objective video analysis tools must be optimised to cope with high data rate in live IPTV video streaming applications. Since audio-visual content are decoded by the user devices such as the set-top box, it would be efficient to integrate relevant evaluation functions as part of the video processing procedures of user devices, should there be sufficient processing capacity.

6 Conclusion

This chapter introduces a multimodal QoE evaluation framework that is specifically designed for quality assessment of the IPTV service in the Lancaster Living Lab. Several subjective and objective evaluation methods are defined and implemented within this framework. Our tests demonstrate the advantages of comprehensive measurement for service evaluation and benchmarking when collaborations between service providers, device manufactures, and network carriers are available. Although the metrics visualised by the framework are relatively complicated for ordinary content consumers, an abstract view can be implemented so that end users can carry out self-diagnosis and receive recommendations if service interruptions are experienced. Future work will focus on further improving the efficiency of measurement components such as objective video analysis and packet inspection to be widely distributed in service networks. The automated detection of quality degradation events and the subsequent diagnostic approaches are also part of the future work. A comprehensive management module (implemented using a number of logics or neural networks) is required to address relationship between layers of measurements as well as the filtering for cases of false positive and false negative results. Moreover, the service metrics derived by the evaluation framework could also be utilised for relevant content and service management mechanisms such as scalable video coding and content-centric networking. Overall, the topic of quality evaluation in Peer-to-Peer IPTV services is complicated, and the work reported here did not address all the issues and challenges. Future work includes the development of new network measurement techniques for efficient P2P traffic monitoring, and procedures for objectively diagnosing IPTV services by analysis of cross-layer measurements.

Acknowledgements. The work presented in this paper is supported by the UK FIRM project (Framework for Innovation and Research at MediaCityUK), grant number EP/H003738/1, and the European Commission within the FP7 P2P-Next project, number FP7-ICT-216217.

References

1. Application layer reliability solutions for IPTV services. ITU-T FG Working document IPTV-ID-0097, ITU (2006)
2. Perceptual evaluation of video quality. Technical report, OPTICOM (2007), http://www.opticom.de/download/SpecSheet_PEVQ_08-03-13.pdf (accessed on July 20, 2011)
3. Final report of VQEG's Multimedia phase I validation test. Video Quality Experts Group (2008), <http://www.its.bldrdoc.gov/vqeg/>
4. Quality of experience requirements for IPTV services. ITU-T FG IPTV Recommendation G.1080, ITU (2008)
5. Alvarez, A., Cabrero, S., Paneda, X.G., Garcia, R., Melendi, D., Orea, R.: A flexible QoE framework for video streaming services. In: Proc. IEEE GLOBECOM Workshops (GC Wkshps), pp. 1226–1230 (2011)

6. Gardikis, G., Boula, L., Xilouris, G., Kourtis, A., Pallis, E., Sidibe, M., Negru, D.: Cross-layer monitoring in IPTV networks 50(7), 76–84 (2012)
7. Klaue, J., Rathke, B., Wolisz, A.: Evalvid - A framework for video transmission and quality evaluation. In: Proceedings of 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Urbana, Illinois, USA, pp. 255–272 (2003)
8. Mol, J., Pouwelse, J., Meulpolder, M., Epema, D., Sips, H.: Give-to-get: Free-riding-resilient video-on-demand in P2P systems. In: Proceeding of the 15th SPIE/ACM Multimedia Computing and Networking, MMCN 2008 (2008)
9. Mu, M., Cerqueira, E., Boavida, F., Mauthe, A.: Quality of experience management framework for real-time multimedia applications. *International Journal of Internet Protocol Technology* 4(1), 54–64 (2009)
10. Mu, M., Gostner, R., Mauthe, A., Garcia, F., Tyson, G.: Visibility of individual packet loss on H.264 encoded video stream: A user study on the impact of packet loss on perceived video quality. In: Proceedings of Sixteenth Annual Multimedia Computing and Networking (MMCN 2009), San Jose, California, USA, ACM (2009)
11. Mu, M., Ishmael, J., Mitchell, K., Race, N.: Multimodal qoe evaluation in p2p-based iptv systems. *ACM Multimedia* (2011)
12. Mu, M., Mauthe, A., Haley, R., Garcia, F.: Discrete quality assessment in IPTV content distribution networks. *Elsevier Journal of Signal Processing: Image Communication* (2011)
13. Mu, M., Romaniak, P., Mauthe, A., Leszczuk, M., Janowski, L., Cerqueira, E.: Framework for the integrated video quality assessment. *Multimedia Tools and Applications*, pp. 1–31, 10.1007/s11042-011-0946-3
14. Pinson, M., Wolf, S.: Comparing subjective video quality testing methodologies. In: Proceedings of SPIE Video Communications and Image Processing Conference, Lugano, Switzerland. SPIE (2003)
15. Rahrer, T., Fiandra, R., Wright, S.: Triple-play services quality of experience (QoE) requirements and mechanisms - For architecture and transport. Technical Report TR-126, Architecture & Transport Working Group, Broadband Forum (2006)
16. Seferidis, V., Ghanbari, M., Pearson, D.E.: Forgiveness effect in subjective assessment of packet video. *IET Journal of Electronics Letters* 28(21), 2013–2014 (1992)
17. ur Rehman Laghari, K., Pham, T.T., Nguyen, H., Crespi, N.: Qom: A new quality of experience framework for multimedia services. In: Proc. IEEE Symp. Computers and Communications (ISCC), pp. 000851–000856 (2012)
18. Wang, Z., Sheikh, H.R., Bovik, A.C.: No-reference perceptual quality assessment of JPEG compressed images. In: Proceedings of IEEE International Conference on Image Processing. IEEE (2002)
19. Wolf, S., Pinson, M.H.: Spatial-temporal distortion metrics for in-service quality monitoring of any digital video system. In: Proc. SPIE International Symposium on Voice, Video, and Data Communications, vol. 3845, pp. 266–277. SPIE (1999)
20. You, F., Zhang, W., Xiao, J.: Packet loss pattern and parametric video quality model for IPTV. In: Proceedings of the 2009 Eighth IEEE/ACIS International Conference on Computer and Information Science, pp. 824–828. IEEE Computer Society (2009)

Cross-Layer FEC-Based Mechanism for Packet Loss Resilient Video Transmission

Roger Immich^{1,*}, Eduardo Cerqueira^{2,*}, and Marilia Curado¹

¹ Department of Informatics Engineering, University of Coimbra
Pinhal de Marrocos, 3030-290, Coimbra, Portugal
{immich,marilia}@dei.uc.pt

² Faculty of Computer Engineering, Federal University of Para
Av. Augusto Correa, 01, 66.075-110, Para, Brazil
cerqueira@ufpa.br

Abstract. Real-time video transmission over wireless networks is now a part of the daily life of users, since it is the vehicle that delivers a wide range of information. The challenge of dealing with the fluctuating bandwidth, scarce resources and time-varying error levels of these networks, reveals the need for packet-loss resilient video transport. Given these conditions, Forward Error Correction (FEC) approaches are desired to ensure the delivery of video services for wireless users with Quality of Experience (QoE) assurance. This work proposes a Cross-layer Video-Aware FEC-based mechanism with Unequal Error Protection (UEP) scheme for packet loss resilient video transmission in wireless networks, which can increase user satisfaction and improve the use of resources. The advantages and disadvantages of the developed mechanism are highlighted through simulations and assessed by means of both subjective and objective QoE metrics.

Keywords: Forward Error Correction (FEC), Video-aware FEC, QoE, Cross-layer, Unequal Error Protection (UEP).

1 Introduction

At present, it is becoming increasingly common to adopt multi-hop wireless networks for different purposes, including a wide range of real-time video services, including streaming. Video streaming has been used by many companies as a part of a business drive to increase productivity, improve collaboration, reduce costs, and streamline and optimize business operations. Following the same trend, non-professional users are producing, sharing and accessing thousands of videos by using both wired and wireless systems. As an example, in October 2011 more than 200 billion videos were viewed online in only one month [1]. This figure means that, on average, each person on the planet has to watch around one online video per day.

* CNPq Fellow - Brazil.

However, new mechanisms for increasing the transmission quality are required to support the growth of video traffic. Additionally, video quality may be affected by several factors, some of which are owing to the video characteristics, such as codec type, bitrate, format and the length of the Group of Pictures (GoP) and, even, the content/genre of the video [2]. A further factor is that the perceptual quality is not affected by all the packets in the same way, because there is a link between the packet content and the impact it has on the user's perception of video quality. When this is taken into account, the most important information should be best protected, and thus encourage the conception and use of the Unequal Error Protection scheme (UEP). The video content also plays an important role during the transmission. Videos with a small degree of movement and few details tend to be more resilient to packet loss. In contrast, videos with high levels of detail and movement are more susceptible to losses and the flaws will be more noticeable [3].

Quality of Experience (QoE) metrics are used to assess the level of the video impairments and can be defined in terms of how users perceive the quality of an application or service [4]. Provided that most of the video services are real-time applications, they need a steady and continuous flow of packets, which can be affected by a number of factors specially in wireless environments. The channel conditions in these networks can suddenly change over time due to noise, co-channel interference, multipath fading, and also, the mobile host movement [5]. Despite the problems outlined above, Wireless Mesh Network (WMN) provides a cost-efficient way of distributing broadband Internet access. Another advantage is its flexibility and reliability for a large set of applications in a wide coverage area [6][7]. Nevertheless, one of the major challenges in WMN is how to distribute the available bandwidth fairly among the requesting nodes to support real-time video traffic [8]. For this reason, it is important to optimize the resource usage and thus avoid congestion periods and a high packet loss rate, particularly in resource-consuming applications, such as video streaming.

The adoption of adjustable data protection approaches is of crucial importance to enhance video transmission, and provide both good perceived video quality and low network overhead. Forward Error Correction (FEC) schemes have been used successfully in real-time systems [9]. FEC allows robust video transmission through redundant data sent along with the original set. As a result, if some information is lost, the original data can be reconstructed through the redundant information [10]. However, as mentioned earlier, the resources might be limited and unfairly distributed as well. An adjustable FEC-based mechanism must use UEP schemes to reduce the amount of redundant information. In this approach, the amount of redundancy is chosen in accordance with the relevance of the protected data, and thus give better protection to the most important video details.

This chapter describes a cross-layer and **VI**dEo-a**W**are FEC-based mechanism (ViewFEC) for packet loss resilient video transmission with UEP. The objective is to strengthen video transmissions, while increasing user satisfaction and improving the usage of wireless resources. Owing to these factors, the use of video-aware FEC-based mechanisms is suitable to transmit videos with better

quality, although it needs additional bandwidth to send the redundant information data. ViewFEC is a novel adaptive mechanism that overcomes this problem by dynamically configuring itself according to the video characteristics and user perception of quality. Using this process only the more sensitive data sets will carry an unequal amount of redundant information, thus maintaining a good video quality and saving resources. The mechanism described here was assessed through simulations with real video sequences, using subjective and objective QoE metrics.

The remainder of this chapter is structured as follows. The related work is shown in Section 2. Section 3 describes ViewFEC and its evaluation is demonstrated in Section 4. Conclusions and future works are summarized in Section 5.

2 Related Work

A wide range of techniques have been proposed to improve the quality of video over wireless networks. The Adaptive Cross-Layer FEC (ACFEC) mechanism uses packet-level error correction at the MAC layer [11]. All the video packets are monitored and, depending on the extent of the losses, the number of FEC recovery packets is either increased or decreased. Nevertheless, no assessment of the network overhead has been conducted, which is very important to assure a fair usage of resources. This approach also does not take account of the video content, and it is well-known that this information has a direct influence on the packet loss resilience of the video [12]. In theory, the ACFEC appears to be a viable solution when the network is healthy and there is sporadic packet loss. However, when congestion occurs, this mechanism will lead to an incremental rise in the number of redundancy packets, which will increase the congestion and cause an unfair distribution of resources.

Another technique that is employed to strengthen the video transmission quality is carried out through a FEC- and retransmission-based adaptive source-channel rate control [13] scheme. The main purpose of this scheme is to ensure the video has playback continuity under uncertain channel variations, as a means of avoiding unneeded FEC redundancy. Video characteristics, such as content and frame type, were not considered in this proposal. In addition, this approach does not use QoE metrics to assess the video quality, as it relies on packet loss levels to predict the QoE values, and finally, it does not measure the network overhead that has been introduced.

A mechanism to improve video transmission over local area wireless networks which use real-time FEC redundancy adjustment and transmission rate was proposed in [14]. To perform the FEC adjustment and transmission rate, all the receivers have to send the network state information to the Access Point (AP) periodically. This mechanism uses application level FEC and pre-encoded video sequences, which means that it will need multiple pre-encoded videos with different bit rates and FEC rate. After this, the system has to switch between different bit streams in real-time. The need for a larger number of pre-processed videos reduces the applicability of this solution in real systems. At the same

time, it also requires a high processing power and storage space, because it has to encode the same video multiple times using distinct bit rates and FEC redundancy data.

A proposal to enhance video streaming transmission using concurrent multi-path and path interleaving was proposed in [15]. This scheme also uses a dynamic FEC block size which is adapted to the average packet loss rate for each path, allowing concurrent interleaved data to be sent, with FEC protection, over multiple paths. This solution only takes into account the network parameters, and leaves out video characteristics, such as codec and frame type, GoP length, motion and complexity. Following the same pattern as the studies outlined above, no attempt is made to measure the network overhead. Due to the factors mentioned earlier, the ViewFEC mechanism aims to enhance the video transmission quality without adding unnecessary network overhead.

3 Video-Aware FEC-Based Mechanism

In the light of the open issues mentioned earlier, this study describes the Video-aware FEC-based Mechanism (ViewFEC) for resilient packet loss video transmission with UEP to enhance video transmission over wireless networks. In ViewFEC, decisions are made at the network layer by resorting to two modules, the **C**luster **A**nalysis **K**nowledge **b**as**E** (CAKE) and the **C**ross-**L**Ayer **i**n**f**or**M**ation (CLAM). The decision-making process at the network layer provides better deployment flexibility, and allows the ViewFEC mechanism to be implemented at access points, routers or in the video server. The analysis of the information obtained from these two modules enables the ViewFEC mechanism to estimate the optimal redundancy ratio necessary to sustain a good video quality, without adding unnecessary network overhead.

The ViewFEC mechanism is depicted in Figure 1. There are 3 distinct stages. In the first stage, through information fetched from CAKE and CLAM, our mechanism identifies several key video characteristics such as motion and complexity levels, as well as GoP length. In the second stage, further details about the video sequence are gathered, namely the type and relative position of the frames within its GoP. The construction of the FEC blocks and the UEP redundancy assignment take place in the third stage. Further details of each module are described later.

CLAM is one of the main modules of ViewFEC and supports the functions that will define the required amount of redundancy needed to maintain a good video quality. These functions are implemented using cross-layer techniques, accessing information of the application layer, such as the video characteristics, from the network layer, where the module was deployed. The objective of the first function is to identify the GoP length. As previously discussed, when the GoP length is larger, the packet loss has a greater influence on the video impairments. This happens partially because, a new I-Frame that is needed to fix the error, will take longer to arrive. Another important remark is that video sequences with high level of spatial complexity tend to have larger I-Frames in

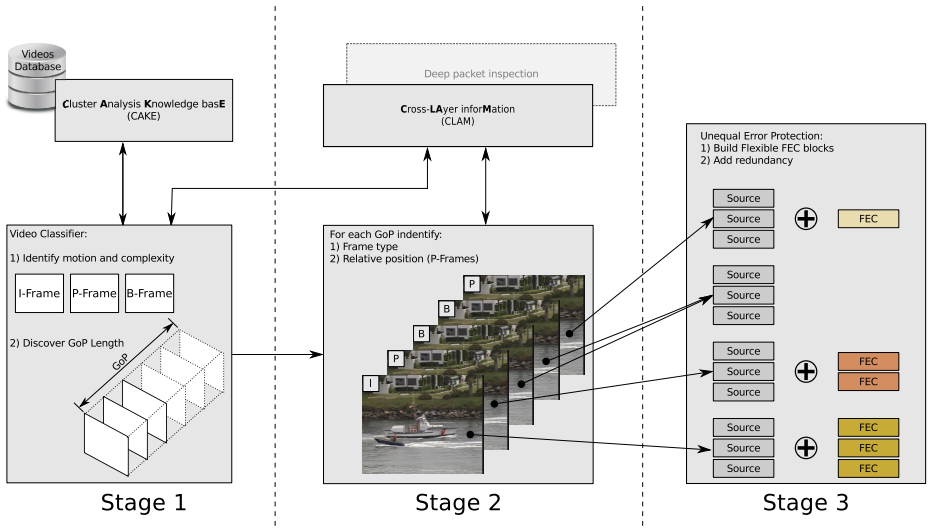


Fig. 1. ViewFEC stages

relation to P- and B-Frames. On the other hand, videos with higher temporal activities tend to have larger P- and B-Frames. Larger frames means that more network packets will be required to carry their data, increasing the chance of these packets being lost. Thus, these packets need more redundancy. Incidentally, as the GoP length is increased, the size of both B-, and especially P-Frames, also increases. In consequence, P-Frames close to the beginning of the GoP take on greater importance regarding the video quality. The role of the second function is to identify the frame type. Different frame types need distinct amounts of redundancy. For example, the loss of an I-Frame will cause more impairments than the loss of a P-Frame, and hence, the loss of a P-Frame will be worse than the loss of a B-Frame. The task of the last function is to identify and compute the relative position of P-Frames inside the GoP. P-Frames that are closer to the beginning of the GoP have more impact, if lost, than those close to the end, and as a result, need more redundancy packets. The combined use of these functions enables ViewFEC to enhance the video quality transmission without adding unnecessary network overhead, and thus, support a higher number of simultaneous users sharing the same wireless resource.

The ViewFEC has a flexible structure, which makes it possible to change the modules to obtain the desired behaviour. When it is not feasible to use a cross-layer design to obtain application layer information, e.g. in a router device, the CLAM module can be exchanged for one that uses another technique to obtain the desired information, for instance, packet and deep packet inspection. By analyzing the packet header of some of the protocols, such as User Datagram Protocol (UDP), Real-time Transport Protocol (RTP) and Transport Stream (TS), it is possible to discover information about codec type and coding parameters, among others. On the other hand, the video content information can only be accessed through deep packet inspection.

The CAKE module optimizes the video transmissions (Figure 1 - Stage 1) through the use of a database with video motion and complexity which is built off-line (before the video distribution). The classification of the video motion and complexity levels needed to create the database is achieved by performing a hierarchical clustering with Euclidean distance. This is a statistical method of partitioning data into groups that are as homogeneous as possible. The video sequences are clustered according to the size of the I-, P- and B-Frames, because they tend to have similar motion activity. This operation only has to be performed once during the setup phase of the mechanism. Afterwards, when the mechanism is running, the relationship between the database information and the videos that are being transmitted in real-time is used to determine a couple of video characteristics, namely motion activity and complexity levels.

The video sequences of the experiments were chosen in compliance with the recommendations of the Video Quality Experts Group (VQEG) [16] and International Telecommunication Union (ITU) [17]. A total of 20 different videos were assessed. Ten of them were used to assemble the database and another set of ten was used to evaluate the ViewFEC mechanism. While remaining in compliance with the recommendations, the videos cover different distortions and content, since they are representative of regular viewing material. As well as this, these sequences contain distinct temporal and spatial details, luminance stress, and still and cut scenes (see Section 4 for more details about the videos).

Video motion and complexity are commonly classified in three categories, namely low, medium and high [3][12] (see Figure 2 at linkage distance (ld) 1). Nevertheless, throughout the experiments, videos with both medium and high complexities behaved roughly the same. Because of this, the linkage distance of our cluster analysis algorithm was chosen to only produce the clusters (Figure 2 at ld 2). This mechanism also employs the Ward method which seeks to reduce the sum of squares between the samples inside the cluster, which better reflects our findings.

The relationship between motion and complexity levels, and frame size (in bytes) and frame position, is shown in Figure 3. This diagram depicts two video sequences - Mobile (A) and Akiyo (B) - each from a different cluster. Only the first GoP of each video was considered, which made it easier to visualize the results. The Mobile sequence has uninterrupted scene modification and a wide-angled camera, and thus, high motion and complexity levels. For this reason, the video has larger frames and greater differences in size between P- and B-Frames, as shown by Figure 3-A. In contrast, the Akiyo video only has a small moving region of interest, just most of the face and shoulders, and also a static background. As a result, it has low motion and complexity levels leading to a smaller difference in size between P- and B-Frames, as depicted by Figure 3-B.

Additionally, the same Figure 3 shows an assessment of the Structural Similarity (SSIM) with frames that have been deliberately discarded. The measurement of this metric is fairly simple, even though it is consistent with the human visual system, and yields good results [18]. The SSIM results were acquired by removing the frame which occupied that position, i.e. the first SSIM value was

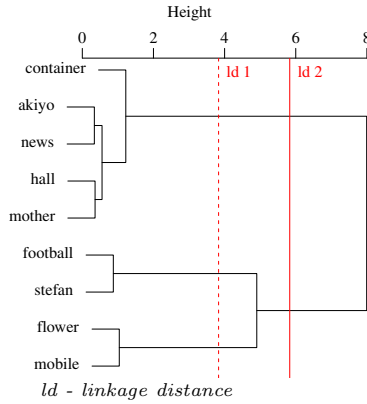


Fig. 2. Cluster Dendrogram

calculated without the first frame, and so forth. It is clear on the basis of these findings that, apart from the fact that in the Mobile video, I- and P- frames have greater significance, the frames closest to the beginning of the GoP also have more impact on QoE video quality when discarded. As expected, the Akiyo sequence behaves differently. It has lower motion and complexity levels being more resilient to packet loss and achieving higher SSIM values [3].

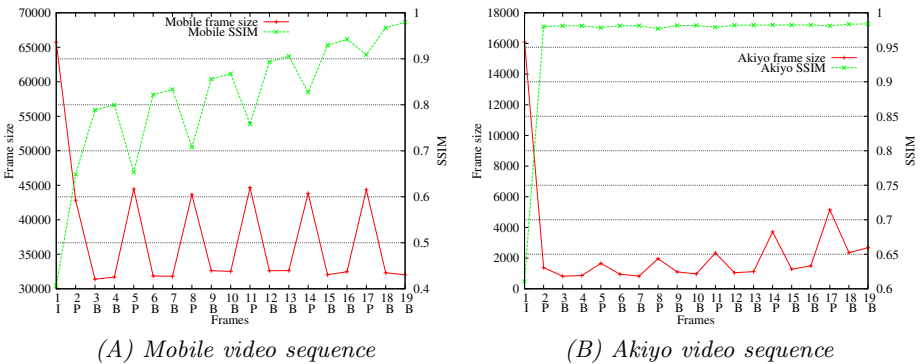


Fig. 3. Frame size x QoE (SSIM)

The CAKE module is aware of these video characteristics and can determine the motion activity and complexity levels of each GoP that are being transmitted. The GoP length is another important factor, which was fetched from the CLAM module. Despite the fact that the GoP length remains the same, all these parameters are assigned GoP by GoP (Figure 1 - Stage 1). This is done because it is possible to have different motion and complexity levels inside the same video

sequence, as expected for Internet videos. The next step (Stage 2) is responsible for retrieving details of the frame type and relative frame position inside the GoP (for P-Frames) from the CLAM module. By the aid of these details, our mechanism will be able to correctly identify the video characteristics needed to configure the amount of redundancy in the next stage.

In the last step (Stage 3), the amount of redundancy needed is calculated in accordance with the details obtained from the previous stages. This tailored amount of redundancy is used to optimally adjust the FEC scheme. In these experiments, a Reed-Solomon (RS) code was used as an erasure code, because it offers less complexity, and consequently achieves a better performance for real-time services [19], but any other alternative scheme could be used. A RS code consists of n , s , and h elements. The total block size, including the redundancy data, is represented by n , and s indicates the original data set size, therefore the parity code is (n, s) . Finally, the parameter h defines the amount of redundancy, which could also be represented as $h = n - s$. Before the original data set s can be restored, at least $(n - h)$ packets have to arrive successfully. The recovery rate can be expressed as h/n or $(n - s)/n$, which means that the robustness to losses is given by the size of h .

The ViewFEC settles the parity code in real-time. In other words, both n and h parameters, are adjusted at Stage 3, based on video characteristics found at Stages 1 and 2, obtained from the CAKE and CLAM modules, respectively. The first parameter of the parity code, n , is used to build the Flexible FEC Block (FFBlock) scheme. This scheme involves dividing the I- and P-Frames into groups of packets, allowing each group to have an individual redundancy data size. This individual size is defined by the second parameter, h , and provides a tailored amount of redundancy for each FFBlock. Hence, rather than using a single redundancy amount to all the frames and video sequences, the ViewFEC mechanism uses an adjustable amount, since it is capable of yielding good results in different network conditions and also supporting a wide range of video characteristics.

The adjustable amount of redundancy data assigned by ViewFEC is the outcome of the joint evaluation of the frame type and position inside the GoP as well as the video motion and complexity levels. By adopting this procedure, we are able to infer the spatio-temporal video characteristics, and as a result, to choose the optimal redundancy amount, h , for each FFBlock. Owing to this, the ViewFEC mechanism is able to achieve better video quality, and has the further advantage of reducing the amount of data that needs to be sent through the network, decreasing the overhead and providing a reasonable usage of wireless resources. This is a very important achievement, because as the network grows larger, the number of concurrent transmissions is increasing with it, and this may cause serious interference problems. The situation gets even worse if we add more overhead due to redundant information. This means that, if the overhead is reduced, a larger number of users will be able to receive more videos with better quality, thus boosting the overall capabilities of the system.

A pseudo-code of the ViewFEC operation is shown in Figure 4. This illustrates how the GoP length and motion detection are performed, and also, the steps taken to assign a tailored amount of redundancy. The algorithm starts with a loop, at line 1, that passes through all the GoPs in a video sequence. At line 4, there is a second loop, which is inside the first one, and will walk through all the frames within a GoP, and only apply the redundancy that is needed. The information retrieval from CAKE and CLAM modules occurs through lines 2, 3, 5, and 11. Since the redundancy amount of P-Frames also depends on their relative position inside the GoP, it has to be treated differently from the I-Frames; this difference is noticeable at line 11.

```

01 for each GoP
02   CAKE.getGopMotion(GoP)
03   CLAM.getGopLength(GoP)
04   for each frame
05     case (CLAM.getFrameType(frame))
06       I-Frame:
07         buildFFBlock(frame)
08         addRedundancy(frame)
09         sendFrame(frame)
10       P-Frame:
11         CLAM.getRelativePosition(frame)
12         buildFFBlock(frame)
13         addRedundancy(frame)
14         sendFrame(frame)
15       B-Frame:
16         sendFrame(frame)
17     end case
18   end for
19 end for

```

Fig. 4. ViewFEC pseudo-code

In Equation 1, it is possible to calculate the amount of redundancy added by the ViewFEC mechanism to each GoP (R_{GoP}). FS_i describes the number of packets of the frame that are being transmitted and FT_i holds the frame type, as shown in Equation 2. If $\gamma > 0$, some level of redundant information will be added to that frame. If we have the vector $(\gamma I, \gamma P, \gamma B)$ with elements $(1, 1, 0)$, for example, only I- and P-Frames will receive redundant information. Additionally, if there is a need to further improve the video quality even if this leads to an increased overhead in the network, the elements of the vector could be $(2, 1, 0)$, meaning that the I-Frames would receive twice the amount of redundant information than the P-Frames (assuming that the other parameters were equal). The parameter C_{GoP} in Equation 3 describes the motion and complexity levels. If the mechanism is using two distinct video clusters, it is possible to define the vectors $(\alpha High/Medium, \alpha Low)$, with elements $(1, 0.5)$,

and ($\alpha High, \alpha Medium, \alpha Low$), with elements (1, 0.5, 0.25). In the former, the cluster with high motion and complexity levels would receive twice the amount of redundancy than the cluster with low levels. If more redundancy levels are needed, the latter could be used, which means that three levels of motion and complexity will be addressed, high, medium and low, respectively. RP_i is the last parameter in Equation 1, which defines the relative distance of the P-Frames inside the GoP. As previously mentioned, frames closer to the end of the GoP are likely to receive less redundant information because the impact of packet loss will be smaller than a loss near the beginning of the GoP, specially in video sequences with larger GoP length. The notation used in the equations is shown in Table 1.

Table 1. Adopted Notation

NOTATION	MEANING
R_{GoP}	ViewFEC redundancy amount per GoP
FS_i	Frame size in packets of number i_{th} frame
FT_i	Frame type of number i_{th} frame
C_{GoP}	GoP motion and complexity level
RP_i	Relative position of number i_{th} P-Frame
N_{GoP}	Number of GoPs in the video sequence

$$R_{GoP} = \sum_{i=0}^{GoPLength} \left[FS_i \times FT_i \times C_{GoP} \times \frac{1}{RP_i} \right] \quad (1)$$

$$FT_i = \begin{cases} \gamma > 0, & \text{send frame with redundancy} \\ 0, & \text{frame without redundancy} \end{cases} \quad (2)$$

$$C_{GoP} = \begin{cases} 1, & \text{high motion and complexity} \\ 0 \leq \alpha < 1, & \text{otherwise} \end{cases} \quad (3)$$

To compute the total amount of redundant information within a video sequence, just perform the sum of all the redundant information of each GoP, which is given by R_{GoP} . On the other hand, the average amount of redundant data, \bar{R} , can be found in Equation 4.

$$\bar{R} = \frac{1}{N_{GoP}} \sum_{i=0}^{N_{GoP}} R_{GoP(i)} \quad (4)$$

4 Performance Evaluation and Results

The primary goal of the ViewFEC mechanism is to reduce the unneeded overhead, while maintaining videos with an acceptable level of quality. The evaluation experiments were carried out by using Network Simulation 3 (NS-3) [20]. The evaluation setting comprises nine nodes placed in a grid form (3x3), 90 meters apart

from the closest neighbour. The routing protocol used was Optimized Link State Routing Protocol (OLSR). An 800 kbps Constant Bit Rate (CBR) background traffic was set. Ten different video sequences were employed in the evaluation scenario [21], with Common Intermediate Format (CIF) size (352x288), H.264 codec and 300 Kbps. All the videos have a GoP length of 19:2, meaning that every 19 frames another I-Frame will be placed and after each two B-Frames, there will be one P-Frame. The error concealment method used by the decoder was Frame-Copy, that is, the lost frames will be replaced by the last good one received. The Gilbert-Elliot loss model is used to produce realistic wireless loss patterns and four different packet loss rates were used: 5%, 10%, 15% and 20%.

Three different cases were used for the video transmission protection. The simplest case (1) is without any type of enhancement (Without FEC). The second case (2) adopts a video-aware FEC-based approach (where both I- and P-Frames are protected in an equal way) with a static amount of redundancy set to 80% (Standard FEC). This amount of redundancy showed the best video quality under the highest packet loss rate defined and was achieved on the basis of a set of detailed experiments. Finally, the last case (3) adopts our proposed approach of an adaptive unequal error protection (ViewFEC). Each of these three cases was simulated 20 times (for packet loss rates of 5%, 10%, 15% and 20%), which means that 80 simulations were carried out for each case, resulting in 240 simulations per video. A total of 10 videos were used, resulting in 2400 simulations. Owing to the distinct initial seeds used to generate the random number, each simulation has a different packet loss pattern.

Subjective and objective QoE metrics were used to assess the video quality obtained from the different cases, namely Structural Similarity Metric (SSIM) [18], Video Quality Metric (VQM) [22] (which were adopted because both are the most widely used objective metrics [23]) and Mean Opinion Score (MOS) [24][25]. With the aid of a set of indicators correlated to the user's perception of quality, the objective metrics perform the assessment without human intervention. The quality assessment was conducted by Evalvid [26] and MSU Tool [27].

Figure 5 shows the average number of the SSIM and VQM values for all the video sequences. In the SSIM metric, the values closer to one indicate a better video quality. As Figure 5-A illustrates, with an increase in the packet loss rate, there is a sharp decline in the video quality of sequences that are transmitted without any type of protection mechanism. At the same time, the video sequences that use either type of FEC-based mechanisms, were able to maintain a good quality. Another important factor that should be noted, is that with 5% and 10% of packet loss rates, the video quality of sequences without FEC are, on average, virtually the same. This can be explained by the natural video resiliency to a certain amount of packet loss. Generally speaking, video sequences with low spatial and temporal complexities are more resilient to loss, and achieve better results in the QoE assessment. Other sequences, with high spatial and temporal complexities, had poor results, and despite the similar average, the standard deviation is higher with a packet loss rate of 10%. This means that the obtained QoE assessment values are more distant from each other. Almost the same pattern is found in the VQM values

in Figure 5-B. In this metric, videos with better quality score close to zero. With a packet loss rate of 5% and 10%, the VQM values are also fairly close to each other. This is not as evident as in the SSIM metric because VQM tends to be more rigid when there are video impairments, and yields poor results to videos with fewer flaws. For the same reason, the standard deviation of this metric has a tendency to be higher than the SSIM metric.

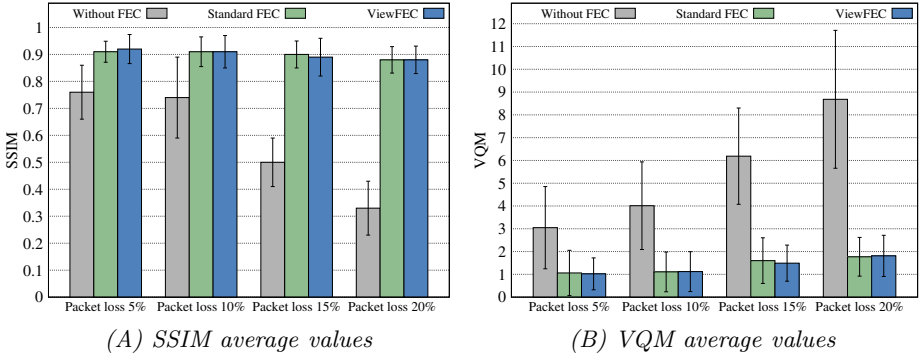


Fig. 5. Average QoE values for all video sequences

Table 2 summarizes the results and shows the improvement of each scenario in percentage terms. The best values for the VQM metric are the smallest, while for SSIM, they are the highest. As expected, both FEC-based mechanisms produce more valuable results when the network has a higher packet loss rate, in our scenario an average of 20%. For example, it was possible to achieve a reduction of over 79% in VQM values, this means ≈ 4.9 times smaller scores. With the SSIM metric, there was an increase of over 166% in the results, meaning ≈ 2.66 times higher values.

Although objective tests can easily assess video quality, they fail to capture aspects of human vision, and thus, subjective evaluations are also required. However this type of assessment tends to be expensive and time-consuming, and in view of this, we decided to select our best-case scenario to perform these experiments. With

Table 2. QoE values and improvement

Packet loss rate	QoE Metric	Without FEC	Video-aware FEC	Video-aware FEC Improvement	ViewFEC	ViewFEC Improvement
Packet loss 5%	VQM	3.05	1.06	↓65.14%	1.02	↓66.48%
	SSIM	0.76	0.91	↑19.74%	0.92	↑21.05%
Packet loss 10%	VQM	4.01	1.11	↓72.36%	1.12	↓72.09%
	SSIM	0.74	0.91	↑22.97%	0.91	↑22.97%
Packet loss 15%	VQM	6.19	1.60	↓74.09%	1.49	↓75.87%
	SSIM	0.50	0.90	↑80.00%	0.89	↑78.00%
Packet loss 20%	VQM	8.68	1.77	↓79.60%	1.81	↓79.14%
	SSIM	0.33	0.88	↑166.67%	0.88	↑166.67%

a packet loss rate of 20%, both Standard and ViewFEC mechanisms achieved better results, and the most significant differences appeared in the videos transmitted without protection mechanism. MOS is one of the most widely used approaches for subjective video evaluation. This follows one of the ITU-T recommendations and uses a predefined quality scale for a group of people scoring video sequences. The MOS scale ranges from 1 to 5, where 5 is the best possible score. A Single Stimulus (SS) method (standard ITU-R BT.500-11 [24]) was chosen because it was a suitable means of carrying out the quality assessment of emerging video applications [28].

The results of the subjective experiments are depicted in Figure 6. The case without FEC has, on average, 2.05 of MOS, which is considered as a poor video quality with annoying impairments. On the other hand, when the FEC-based mechanisms (Standard FEC and ViewFEC) were used, the MOS average values were 4.39 and 4.37, respectively. This indicates that the video quality is between good and excellent, with minor but not annoying impairments, once again, corroborating the objective findings. The different video assessment values, which can be visualized in the results, are due to the unique characteristics of the video sequences. Small differences in motion and complexity levels can influence the obtained values. As a result, it is important to use various types of video when conducting the experiments. This means that a part of the main objective of ViewFEC was achieved, which was to maintain the video quality.

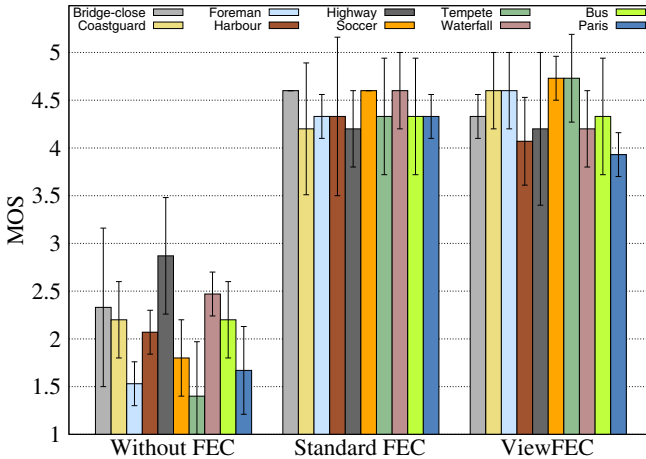


Fig. 6. Average MOS per video sequence with 20% packet loss

Both of the objective and subjective QoE assessments that were employed demonstrated that the ViewFEC mechanism was able to maintain a good video quality in different scenarios. However, another goal of our mechanism is to reduce the network overhead. Due to the limited wireless channel resources, the uneven bandwidth distribution and the interference caused by concurrent transmissions present in WMN, this is a very important issue. In our set up,

the network overhead is given by the summation of the size of all video frames that are transmitted. This allows to measure the specific overhead added by the mechanism. Up to now, neither ViewFEC mechanism nor Standard FEC have been able to adjust the FEC parameters to the state of the network; hence, all packet loss rates have the same FEC overhead. As shown in Figure 7, the network overhead added by the Standard FEC was between 53% and 78%. Conversely, the ViewFEC had considerably less overhead and remains between 34% and 47%. This means that the ViewFEC mechanism imposes, on average, 40% less network overhead, with equal or slightly better video quality, as illustrated by Figures 5 (A and B), and 6.

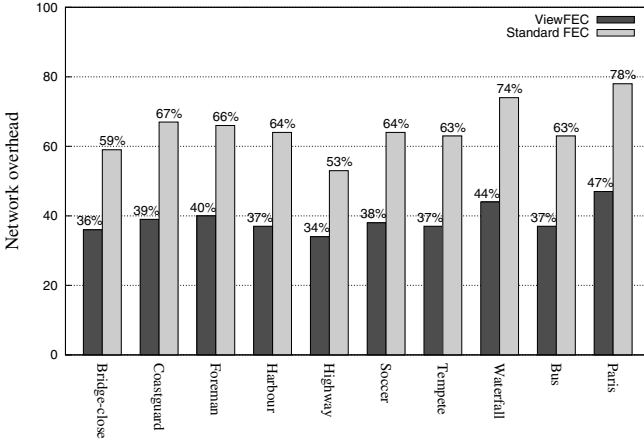


Fig. 7. Network overhead (%)

Owing to a lack of space, the Coastguard video, which is one of the best-case scenarios, was chosen to visualize the results from the viewpoint of the user. Some frames of a video sequence used in the tests, were selected at random and are displayed in Figure 8. In this case, the ViewFEC achieved 4.6 in MOS scale as well as reducing the network overhead, as shown in Figure 6. An improvement of more than 109% was achieved when compared with the video sent without mechanisms to improve the quality (without the FEC scheme, it only reached 2.2 in MOS scale). If compared with the Standard FEC mechanism, which reached 4.2, the ViewFEC still achieves better results with more than 21% improvement in video quality. Furthermore, the standard deviation is considerably smaller, meaning that ViewFEC gets results that are more closely bunched, which indicates a more stable and reliable mechanism.

ViewFEC yielded good results and enabled packet loss resilient video transmission, thus, improving the video quality in the WMN. At this time, the mesh network is only used as a test scenario, and we expect that our mechanism will show its real benefits once it is tailored to these kinds of networks.

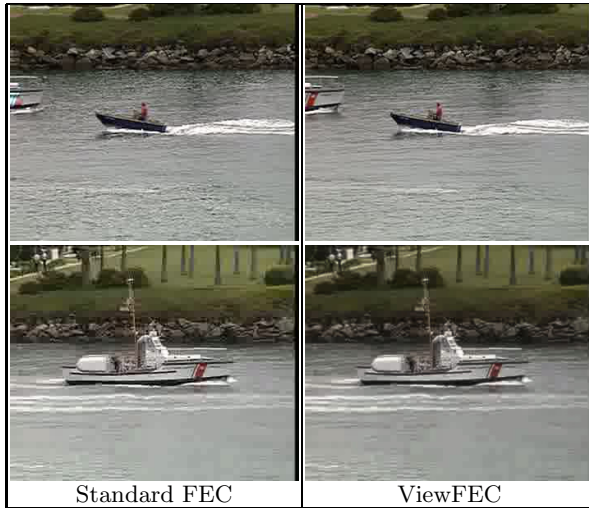


Fig. 8. Frames 10 and 206 of Coastguard video for Standard FEC and ViewFEC

5 Conclusions and Future Works

An effective approach to increase packet loss resilience in video transmission is essential for the growth of video streaming over wireless networks. The Video-Aware FEC-based mechanism for packet loss resilient video transmission provides it with the capacity to enhance video transmission without adding unnecessary network overhead, leading to a better usage of the already scarce wireless resources. A set of controlled experiments was carried out that took into account the different types, complexities, and motions of the video sequences. Network configurations were adopted with different packet loss rates. The simulation results show that the ViewFEC outperforms non-adaptive FEC-based schemes in terms of video quality and in particular, network overhead. The use of Flexible FEC blocks increases the resilience of video transmission to packet losses (by allowing a higher recovery rate), and as a result, achieves better video quality. Video codecs tend to be resilient to a certain amount of packet loss, and because of this, we realized that there was no need to protect all the packets to enhance video quality, especially the information closer to the end of the GoP. The network overhead perceived in the simulations using ViewFEC was between 34% and 47% (on average, 40% less than Standard FEC). As a future work, a systematic variation of the GOP length and structure, as well as the burst length, given by the GE model, will be performed providing a broadest way to validate the results. Also in the next stages, the mechanism will be adjusted to obtain the network state, as a means of enabling it to better adapt to these scenarios as well. Another important point, as previously mentioned, is that through the use of Flexible FEC block, it will be easier to seek out the unique optimization opportunities that WMN offer, i.e. concurrent multipath transmission, network coding and, opportunistic routing.

Acknowledgment. This work was partially funded by the Portuguese Ministry of Science (scholarship contract SFRH/BD/79094/2011) and by FCT UBIQUIMESH project (PTDC/EEATEL/105472/2008).

References

1. comScore, “More than 200 billion online videos viewed globally in october,” comScore inc., Tech. Rep. (2011), http://www.comscore.com/Press_Events/Press_Releases/2011/12/More_than_200_Billion_Online_Videos_Viewed_Globally_in_October
2. Yuan, Y., Cockburn, B., Sikora, T., Mandal, M.: A gop based fec technique for packet based video streaming. In: Conference on Commun (WSEAS). ICCOM 2006, pp. 187–192 (2006)
3. Khan, A., Sun, L., Jammeh, E., Ifeachor, E.: Quality of experience-driven adaptation scheme for video applications over wireless networks. IET Commun. 4(11), 1337–1337 (2010)
4. Piamrat, K., Viho, C., Bonnin, J.-M., Ksentini, A.: Quality of experience measurements for video streaming over wireless networks. In: Third International Conference on Information Technology: New Generations, pp. 1184–1189 (2009)
5. Lindeberg, M., Kristiansen, S., Plagemann, T., Goebel, V.: Challenges and techniques for video streaming over mobile ad hoc networks. Multimedia Systems 17, 51–82 (2011)
6. Zhu, R.: Intelligent rate control for supporting real-time traffic in wlan mesh networks. Journal of Network and Computer Applications 34(5), 1449–1458 (2011)
7. Akyildiz, I., Wang, X.: A survey on wireless mesh networks. IEEE Communications Magazine 43(9), S23–S30 (2005)
8. Liu, T., Liao, W.: Interference-aware qos routing for multi-rate multi-radio multi-channel ieee 802.11 wireless mesh networks. IEEE Transactions on Wireless Communications 8(1), 166–175 (2009)
9. Nafaa, A., Taleb, T., Murphy, L.: Forward error correction strategies for media streaming over wireless networks. IEEE Communications Magazine 46(1), 72–79 (2008)
10. Lee, J.-W., Chen, C.-L., Horng, M.-F., Kuo, Y.-H.: An efficient adaptive fec algorithm for short-term quality control in wireless networks. In: Advanced Communication Technology (ICACT), pp. 1124–1129 (February 2011)
11. Han, L., Park, S., Kang, S.-S., P., H.: An adaptive fec mechanism using cross-layer approach to enhance quality of video transmission over 802.11 wlans. In: THIS, pp. 341–357 (2010)
12. Aguiar, E., Riker, A., Cerqueira, E., Jorge Gomes Abelem, A., Mu, M., Zeadally, S.: Real-time qoe prediction for multimedia applications in wireless mesh networks. In: IEEE 4th Future Multimedia Networking, IEEE FMN 2012 (2012)
13. Hassan, M., Landolsi, T.: A retransmission-based scheme for video streaming over wireless channels. Wirel. Commun. Mob. Comput. 10, 511–521 (2010)
14. Alay, O., Korakis, T., Wang, Y., Panwar, S.S.: Dynamic rate and fec adaptation for video multicast in multi-rate wireless networks. Mobile Networks and Applications 15, 425–434 (2010)
15. Tsai, M.-F., Chilamkurti, N.K., Zeadally, S., Vinel, A.: Concurrent multipath transmission combining forward error correction and path interleaving for video streaming. Comput. Commun. 34, 1125–1136 (2011)

16. Staelens, N., Sedano, I., Barkowsky, M., Janowski, L., Brunnstrom, K., Le Callet, P.: Standardized toolchain and model development for video quality assessment - the mission of the joint effort group in vqeg. In: Quality of Multimedia Experience (QoMEX), pp. 61–66 (September 2011)
17. ITU-T Recommendation J.247, Objective perceptual multimedia video quality measurement in the presence of a full reference, International Telecommunications Union Std. (2008)
18. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* 13(4), 600–612 (2004)
19. Neckebroek, J., Moeneclaey, M., Magli, E.: Comparison of reed-solomon and raptor codes for the protection of video on-demand on the erasure channels. In: Proceedings of Information Theory and its Applications, International Conference, pp. 856–860. IEEE (2010)
20. Weingartner, E., vom Lehn, H., Wehrle, K.: A performance comparison of recent network simulators. In: IEEE Int. Conf. Commun (ICC), pp. 1–5 (2009)
21. Mpeg-4 and h.263 video traces for network performance evaluation, <http://www.tkn.tu-berlin.de/research/trace/trace.html>
22. Pinson, M.H., Wolf, S.: A new standardized method for objectively measuring video quality. *IEEE Trans. on Broadcasting* 50(3), 312–322 (2004)
23. Chikkerur, S., Sundaram, V., Reisslein, M., Karam, L.: Objective video quality assessment methods: A classification, review, and performance comparison. *IEEE Transactions on Broadcasting* 57(2), 165–182 (2011)
24. ITU-R Recommendation BT.500-11, Methodology for the subjective assessment of the quality of television pictures, International Telecommunications Union Std. (2002)
25. ITU-R Recommendation P.910, Subjective video quality assessment methods for multimedia applications, International Telecommunications Union Std. (1999)
26. Klaue, J., Rathke, B., Wolisz, A.: Evalvid - a framework for video transmission and quality evaluation. In: 13th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation, pp. 255–272 (2003)
27. Vatolin, D., Moskin, A., Pretov, O., Trunichkin, N.: Msu video quality measurement tool, http://compression.ru/video/quality_measure/video_measurement_tool.en.html
28. Seshadrinathan, K., Soundararajan, R., Bovik, A., Cormack, L.: Study of subjective and objective quality assessment of video. *IEEE Transactions on Image Processing* 19(6), 1427–1441 (2010)

Approaches for Utility-Based QoE-Driven Optimization of Network Resource Allocation for Multimedia Services

Lea Skorin-Kapov¹, Krunoslav Ivesic¹, Giorgos Aristomenopoulos²,
and Symeon Papavassiliou²

¹FER, University of Zagreb, Croatia

{lea.skorin-kapov, krunoslav.ivesic}@fer.hr

²National Technical University of Athens, Greece

aristome@netmode.ntua.gr, papavass@mail.ntua.gr

Abstract. Taking jointly into account end-user QoE and network resource allocation optimization provides new opportunities for network and service providers in improving user perceived service performance. In this chapter, we discuss state-of-the-art with regards to QoE-driven utility-based optimization of network resource allocation, in particular related to multimedia services. We present two general types of approaches: those which are primarily user-centric and those which are primarily network-centric. Finally, we provide a comparison of the analyzed approaches and present open issues for future research.

Keywords: resource allocation, quality of experience, utility functions, multimedia services, optimization.

1 Introduction

With new and emerging multimedia services imposing increasing resource demands on the network (e.g., video on demand, VoIP, IPTV, interactive networked environments, etc.), a key issue for operators is efficient resource allocation and management (in particular related to wireless networks). Furthermore, the increasing competition in the telecom market powers the everlasting endeavor of both service and network providers to meet end users' requirements in terms of overall perceived service quality and expectations, commonly referred to as the user's Quality of Experience (QoE). Although QoE metrics involve aspects related to subjective user perception and context (e.g., subjective multimedia quality, user expectations based on cost, user environment), also an appropriate mapping to system related characteristics and quantitative network performance parameters such as delay, jitter, loss, and data rate, forming the notion of Quality of Service (QoS), is required.

Consequently, a joint consideration of network resource allocation optimization and QoE provisioning is an upcoming challenge for network providers. In order to devise a QoE-driven resource allocation framework, a two step approach is necessary, providing a clear mapping (quantitative and/or qualitative) of user defined/perceived quality metrics to application parameters (e.g., encoding, frame rate, resolution, content type) and eventually to different network QoS conditions (e.g., delay, packet loss, jitter, bit rate) [30]. Previous work has studied QoE as a mapping to QoS parameters

[11], [13], while a cross-layer approach aiming at network resource allocation optimization focuses on the joint consideration of information collected along different layers, (e.g., application level data, channel quality conditions, etc. [5]).

In order to formalize the correlation between network performance and user perceived quality, utility functions have been defined as a formal mathematical vehicle for expressing user's degree of satisfaction with respect to corresponding multi-criteria service performance [14]. In brief, the concept of utility functions, adopted from economics, provides the means for reflecting in a normalized and transparent way various services' performance prerequisites, users' degree of satisfaction, different types of networks' diverse resources and dissimilar QoS provisioning mechanisms and capabilities, as well as cross-layering information, under common utility-based optimization problems. The goal of QoE provisioning via network QoS-aware resource allocation may thus be restated as to maximize users' aggregated sum of utilities, exploiting Network Utility Maximization (NUM) methods and mechanisms [12].

In this chapter, emphasis is placed on the notion of QoE-driven utility-based optimization of network resource allocation, in particular dealing with multimedia services. We give a comprehensive review and analysis of state-of-the-art solutions applying this concept in different network scenarios and assuming decision-making functionality from different points of view (user-centric, network-centric). Emphasis is placed on recent methodologies that differ and deviate from the traditional point of view of treating Quality of Service (QoS) requirements at the various levels in Internet engineering (application, networking, etc.) creating the need for a more dynamic, interdisciplinary, cross-layer approach to formalize the correlation and impact QoE to QoS that can be engineered and provided within the network. Such approaches escape from the strict bounds of network engineering when studying QoS/QoE, by establishing foundations (e.g., using elements such as dynamic/adaptive network/user utility functions and optimizations) towards creating a framework for interrelating components arising from different points of view (user, provider, operator, engineer).

Following this path, the chapter is organized as follows. Section 2 provides readers with an overview and background on utility-based QoE optimization, including the identification of challenges related to QoE-based resource management. To demonstrate the applicability of the aforementioned methodologies, this chapter will further provide a description of two types of approaches:

- Section 3 will focus on utility-based QoE provisioning in wireless access networks, offering a user-centric approach involving autonomic mobile nodes with enhanced decision-making capabilities towards reacting to mobility and QoS performance related events [6], [4], [3]. Such approaches take explicitly into account user experience and end-user QoE related feedback, while focusing on maximizing users' utility in a real-time manner.
- Section 4 will discuss end-to-end QoE provisioning in the converged NGN, providing a more network-centric decision-making approach with domain-wide QoE optimization being provided in the core network [1], [2], [8], [9], [10], while considering also operator costs and profit. In such approaches, triggers/events driving resource (re)allocation decision-making are commonly detected by network mechanisms.

Finally, the approaches will be compared in more detail and conclusions will be drawn in Section 5, while important open issues for future research will be identified in Section 6.

2 Background on Utility-Based QoE Optimization: Mechanisms and Challenges

2.1 Correlating QoS and QoE

Over the last years the way scientists, engineers, operators and users treat, fulfill and evaluate QoS provisioning has dramatically changed. Considerable efforts have been devoted towards efficient resource utilization, resulting in the evolution from a best effort Internet packet forwarder to a QoS-aware framework, especially for real-time services. Nevertheless, despite the deployment of dynamic resource allocation, traffic shaping and scheduling mechanisms aiming at maintaining services' operation under acceptable networking oriented metrics, such as latency, jitter and packet loss, the final judge of a received multimedia stream still remains the end-user, i.e., a human. In line with the previous, Shenker, in a seminal paper [15] highlighted that "The Internet was designed to meet the needs of users, and so any evaluative criteria must reduce to the following question: how happy does this architecture make the users?" Towards this goal, the concept of utility functions has been adopted and borrowed from economics, allowing the normalization and direct confrontation of users' degree of satisfaction with respect to their multi-criteria service performance. Following this formalism, QoS provisioning problems in wired [12] and wireless networks [16] were designed, modeled and treated via a concrete NUM framework.

However, a human's actual needs and expectations cannot be defined or clearly mapped to strict networking metrics and thresholds, but rather depend on a broader scope of factors. Besides basic QoS networking parameters like bandwidth and jitter, more sophisticated ones include grade of service (GoS) and quality of resilience (QoR) [7], which refer to service connection time and network survivability respectively. Moreover, it also depends on the usage context and intent of usage of the service [29], user role in using a service [27], service content [28], and the users' cultural, socio-economic [21] and psychological state [26], [20]. In [17], for instance, it is shown that if visual factors supplementary to the oral speech are utilized, humans can tolerate higher noise interference levels than in the absence of visual factors.

Consequently, the concept of QoE was developed towards bridging the gap between users' pragmatic needs and provided services' QoS, by elevating users' subjectivity and singularity. While numerous definitions of QoE can be found in literature and standards, a recent definition that has emerged from the EU NoE Qualinet community defines QoE as [32] „*the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user's personality and current state. In the context of communication services, QoE is influenced by service, content, network, device, application, and context of use*“. For instance, Wu *et al.* [30] have portrayed QoE as a user level, sitting on top of application, system and network levels of the protocol stack, with each level relying on underlying quality indicators. As such, application level QoS metrics (e.g., video frame rate, response times) are influenced by network and system QoS, and may further be directly correlated with QoE, noting however that QoE cannot be deduced only from

QoS measurements. However, aiming at users' QoE maximization, though ideal, is not trivial in practice. For instance, while there is an obvious relationship between packet loss and QoE [18], as well as delay and jitter and QoE [13], the authors argue that no clear mapping can be made due to the complexity of the compression and delivery of the services. Moreover, the work in [11] suggests that there is no linear relation between QoE and QoS, but rather an exponential dependency highly related to the data type and content. Towards this goal, various research efforts have mainly concentrated on offline methods with emphasis on determining the factors that influence QoE, measuring and evaluating the corresponding QoE levels and then mapping them to specific network metrics. A typical user-related metric for measuring QoE is the Mean Opinion Score (MOS) [19], which is in general determined from subjective ratings of the content in question by real users. With subjective quality assessment methodologies being time consuming and costly, numerous instrumental, objective methods have been devised aimed at providing quality estimations [31]. However, estimations based solely on metrics such as Peak Signal to Noise Ratio (PSNR) or Mean Square Error (MSE) do not perfectly correlate with perceived quality, e.g., due to the non-linear behavior of the human visual system in the case of video quality assessment [45].

In the next section, generic utility based QoE optimization problems will be drawn, highlighting ways of treatment, challenges and open problems.

2.2 Utility-Based Optimization in the Context of QoE

Following the QoS paradigm shift, and towards enabling a concrete and efficient QoE provisioning framework able of treating the latter multiple and often diverse problem settings, NUM theory has also been exploited, allowing multi-objective subjective performance optimization. To that end, various recent research efforts mainly focus on dynamic schemes that utilize passive or active network monitoring mechanisms and a priori QoE mappings to satisfy the user, relying on existing QoS mechanisms. For instance, a dynamic rate adaptation mechanism is proposed in [22], that maximizes users' cumulative QoE utility-based performance, derived by PESQ (Perceptual Evaluation of Speech Quality) and SSIM (Structural SIMilarity) objective measurements for audio and video services respectively. Moreover, in [23], users' optimal transmission policies in terms of modulation scheme, channel code rate, and share of medium access on wireless networks for various services are determined using a greedy utility-based maximum throughput algorithm. Finally in [21] and [24] a user centric approach to QoE provisioning is considered, where users' viewpoint is taken into account to the overall system QoE optimization problem, either by employing pricing or preference indicators, respectively.

Utility functions have generally been used to specify the relation between relative user satisfaction and consumption of a certain resource. Examples of utility curves corresponding to different types of traffic are portrayed in Fig. 1 (observed by Shenker [15]). *Elastic traffic* can adapt to different network conditions and is generally delay tolerant (e.g., TCP traffic in general, email, file transfer) with decreasing marginal improvement as bandwidth increases (the function is strictly concave). On the other

hand, *discretely adaptive* traffic has strict bandwidth requirements and a sharp loss of utility in between certain thresholds (e.g., audio or video streaming applications operating at discrete codec rates). Commonly audio and video traffic may adapt to different delay/loss values, while bandwidth dropping below a certain intrinsic value causes a sharp performance drop. Corresponding utility functions have been labeled as *adaptive*. In the case of adaptive traffic, the marginal utility of increased bandwidth is small at both high bandwidth and low bandwidth.

Important for making resource allocation decisions is the understanding of the marginal utility change given a change in resource allocation. Reichl et al. [14] have studied the interrelation between QoE metrics and utility functions and argued that logarithmic functions derived based on QoE evaluations occur often in practice.

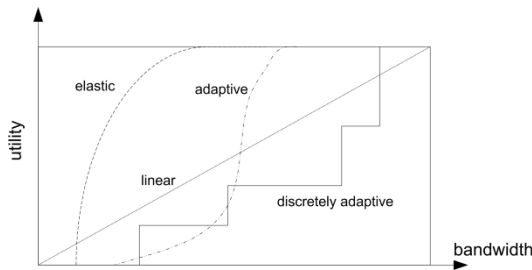


Fig. 1. Example utility curves for different types of traffic

With QoE being of a temporal nature, it can change over time given variations in QoE influencing factors (e.g., network conditions, service state, user preferences, usage context, etc.). Utility functions to be considered for resource allocation may take on different forms during a service's lifetime, for example due to different active media flows, different delay/loss values, change in user terminal (e.g., increased bandwidth and higher resolution will not increase user perceived utility if the user's terminal does not support such a resolution), etc. Aristomenopoulos *et al.* [4] discuss the dynamic service's utility adaptation based on user preferences. On the other hand, in the work done by Ivešić *et al.* [8], the choice of the most suitable utility-functions towards optimal resource allocation is dynamically made based on currently active media flows.

Related to the observations by Reichl et al. regarding applicability of logarithmic laws in user quality perception to QoE of communication services, Thakolsri *et al.* [5] apply utility maximization in the context of QoE-driven resource allocation of wireless video applications. The authors note that certain video quality fluctuations remain unperceived by end users, and exploit this observation in their problem formulation. More specifically, they show how the multiobjective formulation of maximizing average overall quality while minimizing perceived quality fluctuations provides higher average utility for all users as compared to formulations that do not consider quality fluctuation.

In the case of multimodal sessions with multiple media components (e.g., audio and video), different utility functions may correspond to each media component, with overall session utility expressed as some form of a weighted combination. Utility-based multimedia adaptation decision taking has been previously applied in the scope

of the MPEG-21 digital item adaptation standard, and further addressed in the scope of multi-modal media streams by Prangl et al. [33]. A key issue in making multimedia adaptation and resource allocation decisions is consideration of user preferences, e.g., indicating relative importance of individual streams (comprising a single session) such as audio and video (Skorin-Kapov and Matijasevic [10]).

The benefits of QoE-driven resource allocation can range from providing increased end-user/customer satisfaction, to maximizing the number of simultaneous customers (from an operator point of view) while maintaining a certain level of user perceived quality [22]. Different types of generic QoE optimization problems are portrayed in Fig. 2. In a single user case, the focus is on QoE optimization of a given user session taking into account current terminal, network, and service constraints and driven by user QoE estimation methods [3]. On the other hand, multi-user domain-wide QoE optimization problems involve making domain-wide resource allocation decisions across multiple sessions [4], [22], [8]. In practice, the formulation of the objective function for optimizing resource allocation may differ depending on whose interests are being considered (e.g., users' or network operator's). Different examples include: (1) maximizing the (weighted) sum of utility functions across end users, expressed generally as functions of QoS parameters, (2) maximizing the number of "satisfied" users, i.e., with utility above a certain threshold, or (3) maximizing operator profit, by minimizing operator costs. Methods for solving multi-objective optimization may be applied, such as formulation of a composite objective function as the weighted sum of the objectives, or consideration of a Pareto-optimal tradeoff (e.g., between user and network operator or service provider objectives). Possible constraints to be considered include available network and system resources, terminal capabilities, service/user requirements, and cost related constraints (e.g., available user budget). Hence, different actors involved in service delivery (user, network operator, service provider) can be considered in the QoE optimization process, along with their corresponding objectives and constraints.

2.3 Challenges of QoE-Based Resource Management

Numerous challenges may be identified related to the issue of performing QoE-based resource management. The initial concern involves modeling QoE for a given type of service in terms of identifying QoE influence factors and their relationships to QoE metrics. Following specifications of relevant QoE models, monitoring and measurement mechanisms are needed to collect relevant parameters (e.g., related to network performance, user, context, application/service, etc.). Challenges lie in identifying which parameters to collect, where/how to collect data in a scalable manner (e.g., network nodes such as base stations, gateways/routers, application servers; end user terminal), and when to collect data (e.g., before, during, or after service delivery). Finally, mechanisms utilizing collected data for the purpose of QoE-based resource management are needed. Different formulations of QoE optimization problems have been discussed in the previous subsection. Additionally, such mechanisms may involve applying various control mechanisms at the base stations within access

Multimedia sessions comprised of multiple media flows	Maximize <i>QoE</i> for a given user <i>i</i> . <i>QoE</i> expressed as a weighted combination of uni-modal <i>QoE</i> values. Uni-modal <i>QoE</i> values expressed as functions of network resource parameters (at constant values of other influence factors, e.g., usage context, user parameters, service) [10]	Maximize total or average <i>QoE</i> for users $1, \dots, k$ (with possible joint goal of minimizing operator costs or maximizing operator profit). <i>QoE_i</i> , expressed as a weighted combination of uni-modal <i>QoE</i> values for media flows comprising the session of user <i>i</i> . Uni-modal <i>QoE</i> values expressed as functions of network resource parameters (at constant values of other influence factors) [8].
	Sessions considered as single media flow	Maximize <i>QoE</i> for a given user <i>i</i> . <i>QoE</i> values expressed as functions of network resource parameters (at constant values of other influence factors).
	Single user QoE optimization	Multi-user QoE optimization

Fig. 2. Different types of generic QoE optimization problems (for given examples decision variables assumed as network resource parameters)

networks [6], applying policy management rules at the gateways or routers within the core network [22], conducting adjustments at the servers in the service/application [43], content or cloud domains, or the combination thereof [44]. Practical challenges to be considered involve scalability (e.g., support for a large subscriber base), added complexity (e.g., monitoring and parameter collection, optimization calculation, signaling overhead), and additional resulting costs.

Having provided some insight into utility-based optimization approaches and challenges in Section 2, discussing how the existing concept originally drawn from economic theory has been applied in the context of QoE related research, more detailed discussions of certain QoE-driven resource allocation approaches are given in Sections 3 and 4, focusing on user- and network centric decision-making approaches, respectively. While a detailed discussion of meeting the challenges of QoE-based resource management is out of scope for this chapter, we comment on these challenges in the context of approaches discussed in the following Sections.

3 User-Centric Approach to QoE Provisioning in Wireless Networks

The increased interest in QoE provisioning in forthcoming fixed and wireless networks has attracted much attention from the community and various research attempts focusing on the correlation of QoS to QoE have been proposed, mainly focusing on offline MOS objectives tests and network's auto adaptation towards maintaining services' performance under acceptable levels and thresholds [21], [30]. However, experience has been proven to depend on several subjective metrics, including various psychological factors, like mood or the importance of the content to the user. For example in a noisy environment, the presence of subtitles in a video footage can significantly improve users' experience, while also pricing incentives may drive users' behaviour towards tolerating higher interference levels. This implies that no automated mechanism, independently of its complexity and flexibility, is capable of properly dealing with such abstract and often diverse factors. The latter highlights the need

for engaging the end users when testing and evaluating QoE. Staelens *et al.* [41] propose an end-user quality assessment methodology based on full-length DVD movies, which encourages subjects to watch the movie for its content, not for its audiovisual quality, in the same environment as they usually watch television. By providing a questionnaire, feedback can be collected concerning the visual impairments and degradations, which were inserted in the movie. Moreover, Chen *et al.* [42] extend the idea of measurements gathering by proposing Quadrant of Euphoria, a user-friendly web-based crowd-sourcing platform, offering the community end-user subjective QoE assessments for network and multimedia studies.

It is thus imminent that aiming at capturing and determining QoE levels as perceived by the end-user, QoE control should be enabled and conducted at a point in the delivery chain as close to the user as possible, ideally at the end user terminal.

3.1 Autonomics in Wireless Networks

The vast increment of Internet and mobile users, their corresponding services' growing demands on resources and firm QoS expectations, as well as the existence of various available fixed or mobile access network types, assemble the view of the current networking environment, which is mainly characterized by its heterogeneity, multiplicity and complexity. Moreover, within a heterogeneous integrated wireless system, in most cases only the mobile node has the complete view of its own environment, in terms of offered services and their corresponding resource prerequisites, as well as a user's subjective needs and requirements. This becomes even more critical when the available services belong to different providers or even network operators. Therefore, contrary to traditional architectures where network/nodes' performance is monitored and controlled in a centralized way, future wireless networking envisions as its foundation element an autonomic self-optimised wireless node with enhanced capabilities. Such a vision and evolution, supported by the 3GPP LTE Self-Organising / Self-Optimising Networks (SON) initiation [38], proposes the introduction of various self-* functionalities to the end-users allowing them to act and re-act to various events, towards self-optimizing their services' performance. The later presents a promising alternative service oriented paradigm that allows to fully exploit the proliferation of wireless networks, and enhancing users' experience, in terms of improved service performance, QoE, and reduced costs. As such, an autonomic node has the ability and enhanced flexibility to realize a control loop that dynamically, a) exploits locally available information (e.g. types of available services), b) monitors its service performance (e.g. signal strength, connection type), and c) makes optimal service-oriented decisions (e.g. request a HD streaming service) by setting and solving an appropriate optimization problem.

In this scope, NUM theory is envisioned as the enabler to devising network-wide novel autonomic mechanisms capable of optimally driving nodes' behaviour. Moreover, as illustrated in Fig. 3, a generic methodology, extending NUM to the field of autonomics, i.e. Autonomic NUM - ANUM, has been proposed, allowing the design of theoretically-sound autonomic architectures.

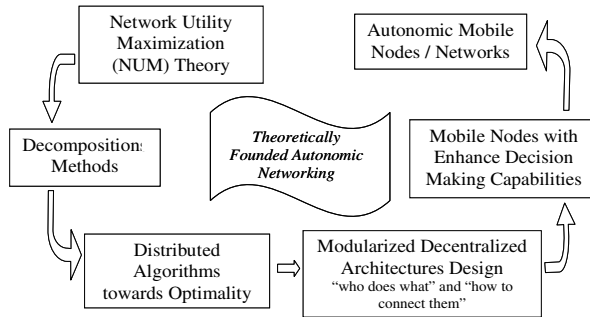


Fig. 3. From Describing to Deriving Autonomic Architectures - A Unified Methodology

3.2 Utility-Based QoE Provisioning in Wireless Networks

Aiming at autonomic QoE provisioning in a multi-user, multi-service heterogeneous wireless network, and considering involved users'/humans' subjectivity, QoE is envisioned as the vehicle that interconnects users/humans, applications and QoS-aware Radio Resource Management (RRM) mechanisms. Aristomenopoulos et al. [6] propose a QoE framework that allows users to dynamically and asynchronously express their (dis)satisfaction with respect to the instantaneous experience of their service quality, as subjectively perceived considering various psychological and environmental influencing factors, at the overall network QoS-aware resource allocation process. Towards this goal, the dynamic adaptation of users' utility functions is proposed, which in turn allows the seamless integration of users' subjectivity in the network utility-based RRM mechanism, enabling cross-layering from the application layer to the MAC layer.

The realization of the aforementioned framework (Fig. 4 [6]) requires a **Graphical User Interface (GUI)** that would a) display and capture user's available options (i.e. various video qualities), and b) present the consequences of his actions in terms of pricing. Upon a user's preference indication, user's service is altered via the **dynamic adaptation** of his/her utility function, and the corresponding RRM mechanism is engaged, towards provisioning the requested resources by solving the resultant **utility-based resource allocation** problem. At this point the feasibility of a user's request as well as its compliance with operator's **policies** can also be introduced towards incorporating for example service performance bounds or fairness among users. Finally, imposed **pricing/billing schemes** correlating a user's QoE-aware behavior, with the corresponding cost of his request can be deployed, providing incentives for users to behave in non-selfish ways that both improve network overall utilization and maximize operators profits.

It is important to note that the proposed approach relies on already existing resource allocation mechanisms, acting complementary upon them, i.e. enabling services' dynamic utility adaptation, thus adding minimal overhead to the overall architecture. This allows the adoption and incorporation of the proposed QoE framework

depends on the form of the utility as well as the service that represents, and should be made in line to the principles:

- A user's parameter $a_i(t)$ should be a step wise function of his preferences (i.e. $a_i(t+1) = a_i(t) \pm A_i \cdot I_i(t)$, where $I_i(t) = 1$ if the user indicates his preference in time t and 0 otherwise, and A_i a fixed predetermined variable).
- A user's parameter $a_i(t)$ and thus, his utility function adaptation, should affect the network's RRM mechanism in such a way to reflect his preferences, e.g., the higher the value of the parameter that determines the unique inflection point of a real-time user's sigmoidal utility (as a function of its achieved rate) the higher his throughput expectations.

The application of the latter methodology for enabling QoE in a diverse heterogeneous wireless network in an autonomic manner requires the extension and proper formulation of traditional utility based RRM mechanisms to flexible, dynamic and adaptive ANUM QoE-aware resource management mechanisms. In the following the modified NUM problems for WLAN and CDMA networks are presented, while a control loop enabling autonomic QoE provisioning is portrayed.

CDMA Cellular Network. Adopting the methodology presented in [39], the following mapping holds, $x_i \equiv p_i$, denoting cell's downlink transmission power allocated to user i and $\sum_{i=1}^N x_i \leq X_{\max} \equiv \sum_{i=1}^N p_i \leq P_{\max}$, denoting cell's overall power constraint. Moreover, the consequent objective function of user's i achieved goodput can be modeled as: $U_i(R_i^{\max}, p_i, \gamma_i) = R_i^{\max} \cdot f_i(p_i, \gamma_i, a_i(t))$, where R_i^{\max} defines user's i maximum downlink transmission rate, γ_i user's i instantaneous signal to noise and interference ratio (SINR) achieved at the mobile terminal, and f_i is a sigmoidal function of the achieved SINR function representing the probability of a successful packet transmission. The latter intra-cell problem can be solved by directly applying the Lagrangian based algorithm in [39].

Wireless LANs. In a similar way, in the case of WLANs [3], x_i denotes the bandwidth allocation from access point (AP) to user i i.e., $x_i \equiv s_i$ and X_{\max} the corresponding AP's maximum effective capacity i.e., C_c^{\max} . Moreover, the consequent bandwidth allocation problem can be modeled in tune to (1) as an optimal contention window assignment problem by initially setting $\sum U_i(s_i, a_i(t))$ under proper effective capacity constraints, where $U_i(s_i, a_i(t))$ is a sigmoidal function representing user's i degree of satisfaction in accordance to his expected allocated effective bandwidth, and finally solved in accordance to [40].

In both cases, functions f and U for CDMA and WLAN systems respectively are sigmoidal functions in general defined as:

$$U_i(x_i, v_i, b_i) = c_i \left\{ \frac{1}{1 + e^{-b_i(x_i - v_i)}} - d \right\} \quad (2)$$

where $c_i = (1 + e^{v_i b_i}) / e^{v_i b_i}$, $d_i = 1 / (1 + e^{v_i b_i})$ and v_i, b_i are two tunable parameters of the sigmoidal function. Parameter v_i determines the function's unique inflection point,

while parameter b_i determines function's steepness. Intuitively, the value of function's inflection point v_i , determines user's i goodput expectations. Moreover, due to the inherent attribute of parameter $a_i(t)$ to imply users' priority among others in being selected for receiving network's resources by the RRM mechanism and thus, attaining larger goodput values, parameter v is selected and exploited by the proposed dynamic QoE framework (i.e. $v_i \equiv a_i(t)$) towards enabling end-users' QoE optimization as follows. When a user experiences low perceived quality of service and requests for a higher service quality then, by decreasing the value of his $a_i(t)$ parameter, in a step-wise manner, user's achieved goodput will be increased, and vice versa. It is important to note that the latter adjustment is performed only when it causes no deterioration of the performance of the rest of the users. In any other case, the user is informed that his request is infeasible.

Having successfully incorporated and properly formalized the QoE provisioning framework to the RRM mechanisms of both CDMA and WLAN networks, by adopting the ANUM principles we define the following control loop, residing at the end-users, enabling autonomic dynamic QoE provisioning in heterogeneous wireless networks.

Quality of Experience Control Loop at the End-User

Step_1. The user constantly monitors his service perceived performance and its corresponding cost via the GUI.

Step_2. If no action is taken, i.e. $I_i(t+1)=0$ go to Step_1. Otherwise, the new $a_i(t+1)$ value is calculated.

Step_3. The service's utility is dynamically adapted and disseminated to the Base Station (or Access Point).

The Base Station (or Access Point) solves the corresponding NUM problem, indicating the feasibility (including policies) of user's request.

Step_4. Resource Management Mechanism allocates the requested resources accordingly. **Go to Step_1**

In case a new user wishes to enter the system, or a new service request occurs, a QoS-aware admission control needs to be performed by the Base Station (or AP). This way, if there exists a feasible resource allocation vector capable of provisioning the newly requested resources, without deteriorating the performance of the already connected ones, then the new user/service is admitted, otherwise is rejected as infeasible.

As already mentioned, the latter dynamic QoE mechanism, via exploiting ANUM theory, is designed and built as a complementary, yet powerful functionality, allowing the seamless integration to existing wireless systems. Specifically, given the operation of the RRM mechanism in each wireless cell, the QoE mechanism acts and reacts on demand, while its decisions will only be evaluated by the RRM mechanism in the next time slot, thus requiring no synchronization. Moreover, it relies only on already existing locally available information, i.e. the perceived quality of the service the node acquires, imposing minimum signaling overhead, while the autonomic nature of the QoE mechanism implies no dependencies on the size and type of the integrated system, thus suggesting it is fully scalable. Finally, indicative numerical results on the performance and effectiveness of the proposed approach reveal the benefits, both from end-users', in terms of increased QoE, and operator's point of view, in terms of increased profits [6].

4 Network-Centric Decision-Making Approach to QoE Management in Converged NGNs

Trends in the development of telecommunication systems indicate the move towards a converged, multi-service all-IP NGN aimed at offering end users integrated services anywhere, anytime [7]. In this Section we discuss approaches whereby QoE-driven resource-allocation stems from a more centralized, operator-centric view of service and resource control in line with ITU-based NGN recommendations [34] and 3GPP specifications [36], [35]. It is important to note that end users are involved in performing QoE/QoS monitoring and reporting, while optimal resource allocation decisions are made in the network. Thus, QoE-driven domain-wide resource allocation may be considered only as a part of the overall QoE management solution provided by a network.

4.1 QoE Management in the NGN Architecture

The ITU-T NGN architecture [34] is based on the concept of independence between the transport stratum and service stratum. In the service stratum, service control functions are based on an IP Multimedia Subsystem (IMS) [36] and support the provisioning of real-time multimedia services, independent of a given access network. The application support functions and service support functions can impact sessions on behalf of services. During session negotiation, QoS requirements are extracted by session control functions and used to issue resource reservation and authorization requests to the resource admission control subsystem (RACS).

A discussion of the challenges in assuring E2E QoE in NGNs is provided by Zhang and Ansari [1]. The authors focus on E2E communications between end users and/or application servers spanning across different access networks (wireless or wireline) and core networks, belonging to multiple operators and based on multiple technologies. Each network performs its own QoE management, and hence actual QoE experienced by an end user will depend on the QoE management mechanisms (or lack thereof) supported by networks traversed along the E2E session path. Feedback provided by end-users is critical in identifying actual user QoE and providing input for such QoE management decisions, which may further drive adaptive transport functions and application configuration parameters. Different end users will exhibit varying preferences and subjective evaluation for the same application or service, and also across different applications/services. The proposed solution stores per-user, per-terminal, and per-service QoE functions in a QoE management block belonging to the NGN service stratum and interacting with the underlying transport stratum to negotiate network-level QoS.

Skorin-Kapov *et al.* [9], [10] have proposed support for enhanced per-session application-level quality matching and optimization functionality by way of a QoS matching and optimization Application Server (QMO AS) included along a session negotiation signaling path (described in further detail in Section 4.2). This concept is illustrated in Fig. 5 in the scope of a generic NGN environment. Communication end points are portrayed as either end users or application servers (AS) offering applications and services. In an actual networking scenario, the QMO AS may be included in a service provider domain as a generic and reusable service capability, supporting

optimized service delivery and controlled service adaptation in light of changing resource availability, user preferences, or service requirements. A business model is assumed whereby the SP is responsible for coordinating the quality negotiation process, while relying on the services of sub-providers (e.g., 3rd party application/service providers, network providers) in order to secure E2E QoS. Utility mappings for multimedia services (comprised of multiple media components) are specified in a *service profile* and signaled from an application server AS to the core network, or retrieved from a service profile repository. Thus, the *service profile* specifies the service resource requirement (as related to service configuration parameters such as e.g., type of media flows, encodings, resolution, etc.). Service requirements are matched with signaled (or retrieved) user parameters (preferences, requirements, capabilities) specified in a *user profile*, network resource availability, and operator policy when calculating optimal resource allocation requests for a given session. Parameters contained in both service and user profiles represent important QoE influence factors to be taken into account when optimizing QoE. Furthermore, the QMO AS may implement interfaces to additional information sources to retrieve additional contextual data, e.g., user location or charging data, to be included in the QoE optimization decision-making process. A further explanation of the QMO AS, and how its functionality may be utilized as input for the purpose of QoE-driven resource allocation (Ivešić et al. [8]), is given in Section 4.2.

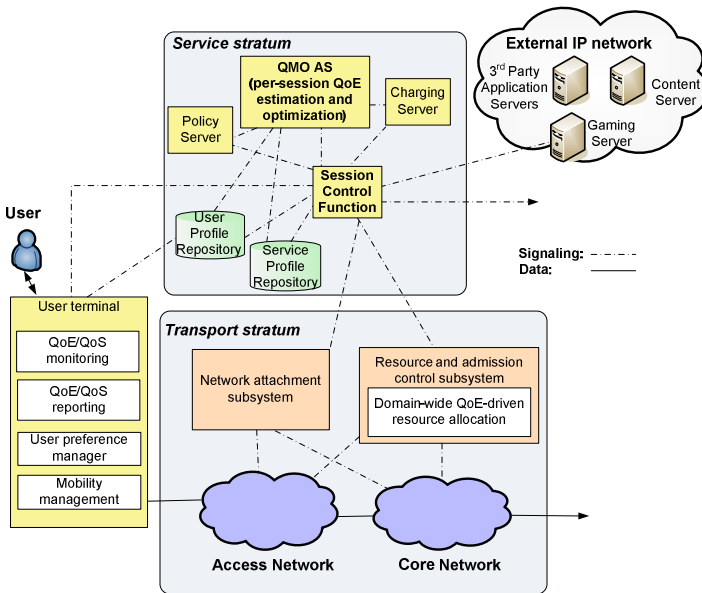


Fig. 5. QoE-driven resource allocation in a generic NGN environment

A related approach to the one previously discussed has been proposed by Volk et al. [25], and studied further by Sterle et al. [2]. Volk et al. have proposed an automated proactive, in-service objective QoE estimation algorithm to be run by a service

enabler AS in the NGN service stratum, based on collection of a comprehensive set of QoE influencing parameters. The proposed algorithm is invoked along the session establishment signaling path, and performs both QoE estimation and QoE maximization calculations. Attempts to maximize QoE are based on making adjustments to identified quality performance indicators. The authors point out the benefits of conducting overall QoE estimation on an AS running in a service delivery environment as opposed to relying only on QoE estimations conducted closer to the user as including: (1) the wide range of quality related information sources available in the network (e.g., databases providing information regarding user profiles and personalized communication scenarios, service profiles, QoS monitoring, charging related information, operator policy) reachable via standardized protocols (e.g., SIP, Diameter), and (2) the potential for proactive in-service quality assurance and control.

4.2 QoE-Driven Dynamic Resource Allocation for Adaptive Multimedia Services

In this Section we discuss in further detail the approach studied by Ivešić *et al.* [37], [8] related to QoE-driven resource allocation for adaptive multimedia services. By adaptive services, it is assumed that a service configuration may be varied in various ways (e.g., using different codecs, bit rates, resolution, etc.) in order to address the wide variety of terminal equipment, access networks capabilities, and user preferences. It has been noted that in the case of multimedia services comprised of multiple media components, user preferences regarding the relative importance of different components may vary.

The previously discussed QMO AS (shown in Fig. 5) is invoked at service establishment and gathers input parameters (related to the user profile, service profile, and operator policies). An initial parameter matching process is conducted to determine feasible service configurations, followed by a utility-based optimization process used to determine the optimal service configuration and resource allocation (referred to as the optimal operating point) for the given service session. The resulting operating point is the basis for the optimal service *configuration*, i.e., the specification of flows operating parameters (e.g., frame rate, codec), resource requirements (e.g., bit rate) and a utility value that represents a numerical estimation of the configuration's QoE. Besides the optimal configuration, several suboptimal configurations are calculated and ordered by their decreasing utility value, thus forming a *Media Degradation Path* (MDP). The goal of the MDP is to serve as a “recipe” for controlled service adaptation, achieving maximum utility in light of dynamic conditions. For example, in the case of a user indicating that he/she prefers audio over video for a given audiovisual service, an MDP may be constructed so as to first degrade video quality in light of a decrease in network resource availability, while maintaining high audio quality. Hence, in light of decreased resource availability, a suboptimal configuration can be activated (thus preventing unpredictable degradation of a service). Since the media components of the service are not necessarily all active at the same time, the configurations are grouped by the *service state* they pertain to, whereby the service state refers to a set of service components simultaneously active during a given time period.

Fig. 6 illustrates an example MDP for a 3D virtual world application with the possibility of activating a video stream or an audio chat. The MDP consists of three service states with several corresponding configurations defined for each state. Since states 2 and 3 consist of two service components each, their configurations should ensure that the available resources for virtual world and video or audio are divided according to user preferences in case of activation of a suboptimal configuration. Similarly, in state 1, suboptimal configurations can use smaller levels of details for the virtual world, rather than causing slow download of a virtual world. In this way, the knowledge about the service and the user is encompassed in the MDP.

Calculated per-session MDPs may be passed on to a resource and admission control entity responsible for making domain-wide resource allocation decisions (we note that possible dynamic changes in user preferences signaled by an end user may lead to recalculation of the MDP). Optimal resource allocation among multiple sessions has been formulated as a multi-objective optimization problem with the objectives of maximizing the total utility of all active sessions described by their MDPs, along with operator profit. The regarding problem class is multi-choice multidimensional 0-1 knapsack problem (MMKP). The problem formulation has been given in [8] in the context of the 3GPP Evolved Packet System (EPS), which maps session flows to one of 9 different QoS Class Identifiers (QCIs) – identifiers standardized by the 3GPP that define different types of standardized packet forwarding behavior. (Note that this is only one possible problem formulation, focusing on discrete optimization and assuming a weighted combination of multiple objectives. Examples of other optimization objectives are discussed in Section 2.2.)

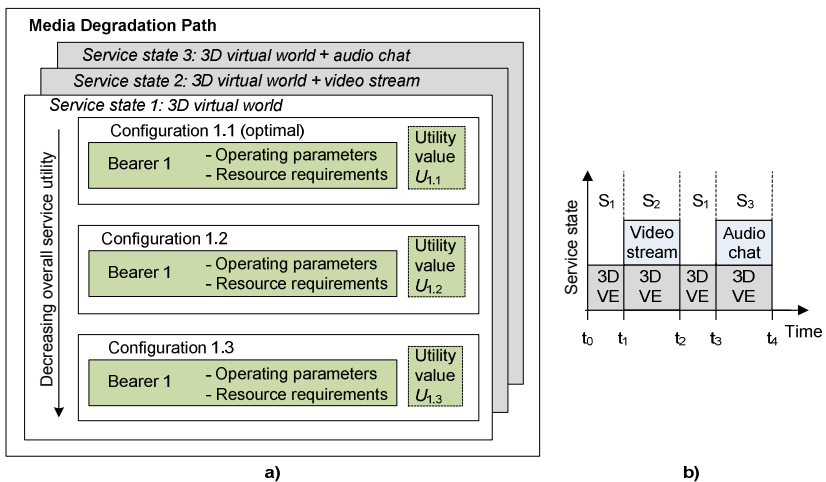


Fig. 6. (a) Example Media Degradation Path, (b) possible service scenario (different service states active at different intervals of service execution)

The formulation is as follows. Let n be the number of sessions, p_u the number of configurations of the currently active state of the session u . Let the configuration i of session u have z_{ui} media flows, such that the flows 1, ..., h_{ui} pertain to downlink and

the flows $h_{ui}+1, \dots, z_{ui}$ pertain to uplink. Let $\mathbf{b}_{ui} = (\mathbf{b}_{ui1}, \dots, \mathbf{b}_{uiz_{ui}})$ be the bandwidth requirements of the configuration i , with $\mathbf{b}_{uij} = (b_{uij1}, \dots, b_{uij9})$ being the vector describing bandwidth requirements of the media component j with regards to each of the 9 QCIs. It is assumed that only a single QCI bandwidth is greater than zero (i.e., j is mapped to a single QCI) while the others are equal to zero. Let $U_n(\mathbf{b}_{ui})$ and $P_n(\mathbf{b}_{ui})$ be the normalized users' utility and operator's profit of a configuration i and w_u, w_{ut} , and w_{pr} weight factors for user u (different weight factors may be assigned to different users, e.g., "premium" and "regular"), users' utility and operator's profit respectively. The normalization is conducted to enable the fair comparison of configurations belonging to different services, by dividing the utilities of all the configuration of the regarding service state from the MDP with the utility of the first configuration (the highest one), and the profits with the profit of the configuration that brings the highest operator's profit (not necessarily the first one). Let B_{kD} and B_{kU} denote the total available bandwidth of QCI k for downlink and uplink respectively. Then, the optimization problem is formulated as:

$$\max \sum_{u=1}^n \sum_{i=1}^{P_u} \left\{ w_u x_{ui} \left[w_{ut} U_n(\mathbf{b}_{ui}) + w_{pr} P_n(\mathbf{b}_{ui}) \right] \right\} \quad (3)$$

such that:

$$\sum_{u=1}^n \sum_{i=1}^{P_u} \sum_{j=1}^{h_{ui}} x_{ui} b_{uijk} \leq B_{kD}, k = 1, \dots, 9 \quad (4)$$

$$\sum_{u=1}^n \sum_{i=1}^{P_u} \sum_{j=h_{ui}+1}^{z_{ui}} x_{ui} b_{uijk} \leq B_{kU}, k = 1, \dots, 9 \quad (5)$$

$$\sum_{i=1}^{P_u} x_{ui} = 1, x_{ui} \in \{0, 1\}, u = 1, \dots, n \quad (6)$$

The solution to the maximization problem is the list of the selected configurations from all active sessions (determining in turn the resources to be allocated to each flow), indicated by the binary variables x_{ui} . Since the MMKP problem is NP-complete, finding the exact solution quickly becomes too time consuming. Therefore, dedicated heuristics may be applied in order to obtain good results in short computational time. Additionally, if a large number of sessions are affected by the optimization process, the resulting signaling overhead used to notify session entities of new configurations would need to be considered. A simulator tool used to evaluate the above given formulation is described in [8].

In the context of the NGN, this approach may be applied in the scope of RACS, or applied at a lower level for optimized resource allocation in a given access network, such as a cell area covered by an eNodeB base station in an LTE network. In the latter case, MDP information would need to be calculated at an application-level and passed to lower level resource allocation mechanisms implemented by a base station. Applicability of the proposed approach in the context of LTE resource scheduling is a current area of research.

5 Discussion and Conclusions

Using as a basis the notion of utility-based QoE-driven resource allocation, we have described two different types of approaches which have been proposed to tackle such problems: primarily user-centric solutions, and primarily network-centric solutions. We summarize key differences as follows:

- A user-centric approach takes explicitly into account the user experience while a network-centric approach can be considered as an implicit way of treating QoE. However, it should be noted that certain network centric approaches also support the collection of end-user QoE-related feedback.
- With regards to optimization objectives, network-centric approaches aim at maximization of global, multi-user QoE while at the same time explicitly minimizing operator costs or maximizing operator profit. A user-centric approach would primarily focus on maximizing all users' utilities by meeting in a real-time manner their individual QoE requirements, with network operation optimality implicitly ensured by proper resource allocation mechanisms.
- Scalability in user-centric approach is ensured by allowing the user device to partially participate in the optimization process, while in network-centric approaches scalability can be achieved by considering aggregated objectives (referring to joint consideration of the objectives of multiple users in a given domain).
- User-centric approaches may require more powerful and smart devices capable of autonomic decision-making related to self-optimization of service performance, while network-centric approaches may work even with more conventional end devices and legacy systems. Furthermore, a user-centric approach would be able to operate in a multi-provider environment without direct involvement of the operators/providers (e.g., a user switching from one provider to another to improve QoE), given that such operations are permitted.
- With regards to triggers/events driving resource (re)allocation decision making, in a user-centric approach such triggers may be considered as coming from individual users through their expression of quality requests, while in a network-centric approach, the network will commonly detect when to perform resource re-allocation (e.g., based on identified network congestion, operator policy, input from charging system, etc.). In the latter case, network-based resource allocation decision making may also be invoked based on detected changes in service requirements (e.g., an existing service is modified with the addition/removal of a media component) and in certain cases based on signaled changes in user preferences.

Considering the applicability of the different approaches, a network-centric approach is closer to the NGN/IMS operator-centric model adopted by telecom providers looking to maintain as much call/session control as possible (related both to QoS and charging), while the Internet community is looking towards a more decentralized network model with intelligence being pushed towards the network edges. Increasingly, however, operators are also looking to incorporate user-centric experience management into their network management solutions.

It is clear that in order to estimate true QoE, quality-related feedback needs to be collected from the network edges, i.e., directly from the end users. Consequently, it

will be necessary for network-centric approaches to ultimately combine notions described as user-centric given that QoE is inherently user-centric. The user perceived QoE related to delivered services, however, will in most cases depend largely on the underlying network performance. With resource allocation decisions being inherently made in the network, an end node capable of making decisions reflecting how to maximize the given end user QoE (e.g., by incorporating dynamic preferences indicated by end users) can provide valuable input for the network decision making process. On the other hand, certain information which may be relevant in making optimal allocation decisions (e.g., operator policy, subscriber data, service priority, network resource availability) may only be available in the network.

In such a case, information related to QoE management needs to be exchanged among different players involved (users, application/service providers, network providers, etc.). End user benefits include improved QoE, service provider benefits include increased user/customer satisfaction, and network provider benefits include reduced costs based on more efficient resource usage together with increased customer satisfaction.

6 Open Research Issues

While it is clear that user and network centric approaches to QoE-driven resource allocation problems differ as described in the previous section, an open research issue would be to provide a more detailed analysis and comparison of the achieved results when solving the resource allocation problem from a user point of view as compared to solving the problem from a network point of view. Such an analysis would involve determining whether the solutions would be very different or if there would be a certain degree of correlation.

Additional possibilities for future research involve combining the key benefits of user and network-centric approaches in order to achieve a scalable QoE-driven resource allocation solution for future networks. Furthermore, with the advent of QoE-related research leading to a better understanding of QoE models and the correlation between numerous QoE influence factors and QoE metrics, new formulations for solving QoE optimization problems (in particular for new and emerging services) will need to be considered.

Acknowledgments. This work was supported in part by the EC in the context of the TMA (COST IC0703) Action. The work done by L. Skorin-Kapov and K. Ivešić was in part supported by the Ministry of Science, Education and Sports of the Republic of Croatia research projects no. 036-0362027-1639 and 071-0362027-2329.

References

1. Zhang, J., Ansari, N.: On Assuring End-to-End QoE in Next Generation Networks: Challenges and a Possible Solution. *IEEE Comm. Mag.* 49(7), 185–191 (2011)
2. Sterle, J., et al.: Application-Based NGN QoE Controller. *IEEE Comm. Mag.* 49(1) (January 2011)

3. Varela, M., Laulajainen, J.-P.: QoE-Driven Mobility Management – Integrating the Users’ Quality Perception into Network-Level Decision Making. In: Proc. of QoMEX 2011, pp. 19–24 (September 2011)
4. Aristomenopoulos, G., Kastrinogiannis, T., Li, Z., Papavassiliou, S.: An Autonomic QoS-centric Architecture for Integrated Heterogeneous Wireless Networks. *Mobile Networks and Applications* 16(4), 490–504 (2011)
5. Thakolsri, S., Kellerer, W., Steinbach, E.: QoE-based Cross-Layer Optimization of Wireless Video With Unperceivable Temporal Video Quality Fluctuation. In: Proc. of ICC 2011, pp. 1–6 (July 2011)
6. Aristomenopoulos, G., Kastrinogiannis, T., Kladanis, V., Karantonis, G., Papavassiliou, S.: A Novel Framework for Dynamic Utility-Based QoE Provisioning in Wireless Networks. In: Proc. of IEEE GLOBECOM 2010, pp. 1–6 (2010)
7. Stankiewicz, R., Jajszczyk, A.: A Survey of QoE Assurance in Converged Networks. *Computer Networks* 55(7), 1459–1473 (2011)
8. Ivešić, K., Matijašević, M., Skorin-Kapov, L.: Simulation Based Evaluation of Dynamic Resource Allocation for Adaptive Multimedia Services. In: Proc. of the 7th International Conference on Network and Service Management (CNSM), pp. 1–8 (October 2011)
9. Skorin-Kapov, L., Mošmondor, M., Dobrijević, O., Matijašević, M.: Application-level QoS Negotiation and Signaling for Advanced Multimedia Services in the IMS. *IEEE Communications Magazine* 45(7), 108–116 (2007)
10. Skorin-Kapov, L., Matijasevic, M.: Modeling of a QoS Matching and Optimization Function for Multimedia Services in the NGN. In: Pfeifer, T., Bellavista, P. (eds.) MMNS 2009. LNCS, vol. 5842, pp. 55–68. Springer, Heidelberg (2009)
11. Fiedler, M., Hossfeld, T., Tran-Gia, P.: A Generic Quantitative Relationship between Quality of Experience and Quality of Service. *IEEE Network* 24(2), 36–41 (2010)
12. Chiang, M., Low, S.H., Calderbank, A.R., Doyle, J.C.: Layering as Optimization Decomposition: A mathematical theory of network architectures. *Proc. of the IEEE* 95(1), 255–312 (2007)
13. Gulliver, S.R., Ghinea, G.: The perceptual influence of multimedia delay and jitter. In: Proc. of the 2007 IEEE International Conf. on Multimedia and Expo., pp. 2214–2217 (2007)
14. Reichl, P., Tuffin, B., Schatz, R.: Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience. *Telecommunication Systems*, 1–14 (June 2011)
15. Shenker, S.: Fundamental design issues for the future Internet. *IEEE Journal on Selected Areas in Communications* 13(7), 1176–1188 (1995)
16. Lee, J.W., Mazumdar, R.R., Shroff, N.B.: Downlink power allocation for multi-class wireless systems. *IEEE/ACM Trans. on Net.* 13(4), 854–867 (2005)
17. Sumbly, W.H., Pollack, I.: Visual contribution to speech intelligibility in noise. *Journal of the Acoustical Society of America* 26(2), 212–215 (1954)
18. Liang, Y.J., Apostolopoulos, J.G., Girod, B.: Analysis of packet loss for compressed video: effect of burst losses and correlation between error frames. *IEEE Transactions on Circuits and Systems for Video Technology* 18(7), 861–874 (2008)
19. International Telecommunication Union, “Definition of Quality of Experience”, ITU-T Delayed Contribution D.197, Source: Nortel Networks, Canada, P. Coverdale (2004)
20. De Moor, K., Joseph, W., Tanghe, E., Ketyko, I., Deryckere, T., Martens, L., De Marez, L.: Linking Users’ Subjective QoE Evaluation to Signal Strength in an IEEE 802.11b/g Wireless LAN Environment. *EURASIP Journal on Wireless Communications and Networking* 2010 (2010), doi:10.1155/2010/541568

21. Reichl, P., Fabini, J., Kurtansky, P.: A Stimulus-Response Mechanism for Charging Enhanced Quality-of-User Experience in Next Generation All-IP Networks. In: Proc of CLAIO 2006, Montevideo, Uruguay (November 2006)
22. Thakolsri, S., Khan, S., Steinbach, E., Kellerer, W.: QoE-Driven Cross-Layer Optimization for High Speed Downlink Packet Access. *Journal of Communications* 4(9), 669–680 (2009), doi:10.4304/jcm.4.9.669-680
23. Khan, S., Duhovnikov, S., Steinbach, E., Kellerer, W.: MOS-Based Multiuser Multi application Cross-Layer Optimization for Mobile Multimedia Communication. *Advances in Multimedia 2007* (2007), doi:10.1155/2007/94918
24. Derbel, H., Agoulmine, N., Salaun, M.: Service Utility Optimization Model Based on User Preferences in Multiservice IP Networks. In: 2007 IEEE Proc. of Globecom Workshops (2007), doi:10.1109/GLOCOMW.2007.4437817
25. Volk, M., Sedlar, U., Kos, A.: An approach to modeling and control of qoe in next generation networks. *IEEE Communications Magazine* (2010)
26. Möller, S., et al.: A Taxonomy of Quality of Service and Quality of Experience of Multimodal Human-Machine Interaction. In: International Workshop on Quality of Multimedia Experience, QoMEx, pp. 7–12 (July 2009)
27. Kilkki, K.: Quality of Experience in Communications Ecosystem. *Journal of Universal Computer Science* 14(5), 615–624 (2008)
28. Gulliver, S., Ghinea, G.: Defining User Perception of Distributed Multimedia Quality. *ACM Transactions on Multimedia Computing, Communications and Applications* 2(4), 241–257 (2006)
29. Wac, K., et al.: Studying the Experience of Mobile Applications Used in Different Contexts of Daily Life. In: Proceedings of ACM SIGCOMM W-MUST (August 2011)
30. Wu, W., et al.: Quality of Experience in Distributed Interactive Multimedia Environments: Toward a Theoretical Framework. In: Proceedings of the 17th ACM International Conference on Multimedia (2009)
31. ITU-T Recommendation G.1011. Reference guide to quality of experience assessment methodologies (June 2010)
32. Qualinet White Paper on Definitions of Quality of Experience (QoE) (May 2012), <http://www.qualinet.eu/>
33. Prangle, M., Szkaliczki, T., Hellwanger, H.: A Framework for Utility-Based Multimedia Adaptation. *IEEE Transactions on Circuits and Systems for Video Technology* 17(6), 719–728 (2007)
34. ITU-T Recommendation Y.2012. Functional requirements and architecture of the NGN release 1 (2006)
35. 3GPP TS 23.203, Policy and Charging Control Architecture, Rel. 11 (2012)
36. 3GPP TS 23.228: IP Multimedia Subsystem (IMS); Stage 2, Release 11 (2012)
37. Ivešić, K., Matijašević, M., Skorin-Kapov, L.: Utility Based Model for Optimized Resource Allocation for Adaptive Multimedia Services. In: Proc. of the 21st PIMRC 2010, pp. 2636–2641 (2010)
38. 3GPP TSG SA, 3GPP TS 32.500. Telecommunication management; Self-Organizing Networks (SON); Concepts and requirements
39. Lee, J., Mazumdar, R., Shroff, N.: Joint resource allocation and base-station assignment for the downlink in CDMA networks. *IEEE/ACM Trans. on Networking* 14, 1–14 (2006)
40. Yang, Y., Wang, J., Kravets, R.: Distributed Optimal Contention Window Control for Elastic Traffic in Single-Cell Wireless LANs. *IEEE/ACM Trans. on Netw.* 15(6), 1373–1386 (2007)

41. Staelens, N., Moens, S., Van den Broeck, W., Marien, I., Vermeulen, B., Lambert, P., Van de Walle, R., Demeester, P.: Assessing Quality of Experience of IPTV and Video on Demand Services in Real-Life Environments. *IEEE Transactions on Broadcasting* 56(4), 458–466 (2010)
42. Chen, K.-T., Chang, C.-J., Wu, C.-C., Chang, Y.-C., Lei, C.-L.: Quadrant of euphoria: a crowdsourcing platform for QoE assessment. *IEEE Network* 24(2), 28–35 (2010)
43. Latre, S., et al.: An Autonomic Architecture for Optimizing QoE in Multimedia Access Networks. *Journal of Computer Networks: The International Journal of Computer and Telecommunications Networking* 53(10), 1587–1602 (2009)
44. El Essaili, A., Zhou, L., Schroeder, D., Steinbach, E., Kellerer, W.: QoE-driven Live and On-demand LTE Uplink Video Transmission. In: *Proc. of the 13th International Workshop on Multimedia Signal Processing (MMSP 2011)*, Hangzhou, China (October 2011)
45. Winkler, S.: *Digital Video Quality: Vision Models and Metrics*. John Wiley & Sons (2005)

Author Index

- Aceto, Giuseppe 28
Aracil, Javier 3
Aristomenopoulos, Giorgos 337
- Biersack, Ernst 264
Botta, Alessio 28
- Callegari, Christian 148
Cerqueira, Eduardo 302, 320
Coluccia, Angelo 148, 202
Curado, Marilia 320
- Dainotti, Alberto 123
D'Alconzo, Alessandro 148, 202
Davy, Alan 28
Donnet, Benoit 44
- Egger, Sebastian 219
Ellens, Wendy 148
- Finamore, Alessandro 123
- García-Dorado, José Luis 3
Georgopoulos, Panagiotis 302
Giordano, Stefano 148
- Höfelfeld, Tobias 219, 264
Hutchison, David 302
- Immich, Roger 320
Ivesic, Krunoslav 337
- Janowski, Lucjan 219
- Knowles, William 302
Krieger, Udo R. 104
- Mandjes, Michel 148, 184
Markovich, Natalia M. 104
Mata, Felipe 3
Mauthe, Andreas 302
Mellia, Marco 123
Meskill, Brian 28
Moreno, Victor 3
Mu, Mu 302
- Pagano, Michele 148
Papavassiliou, Symeon 337
Pepe, Teresa 148
Pescapè, Antonio 123
Plissonneau, Louis 264
- Race, Nicholas 302
Ramos, Javier 3
Ricciato, Fabio 148, 202
Rossi, Dario 123
- Santiago del Río, Pedro M. 3
Schatz, Raimund 219, 264
Shavitt, Yuval 82
Simpson, Steven 302
Skorin-Kapov, Lea 337
- Valenti, Silvio 123
- Zilberman, Noa 82
Żuraniewski, Piotr 148, 184