TECHNISCHE UNIVERSITÄT BERLIN
FAKULTÄT FÜR ELEKTROTECHNIK UND INFORMATIK
LEHRSTUHL FÜR INTELLIGENTE NETZE
UND MANAGEMENT VERTEILTER SYSTEME

Impact of Network Effects on Application Quality

vorgelegt von
Muhammad Amir Mehmood (M.Sc.)

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
- Dr.-Ing. -
genehmigte Dissertation

Promotionsausschuss:

Vorsitzende:	Prof. Dr. Axel Küpper, Technische Universität Berlin
Berichtende:	Prof. Anja Feldmann, Ph. D., Technische Universität Berlin
Berichtender:	Prof. Dr. Steve Uhlig, Queen Mary University of London, UK
Berichtender:	Dr. Walter Willinger, AT&T Labs–Research, USA

Tag der wissenschaftlichen Aussprache: 09.07.2012

Berlin 2012
D83

Ich versichere von Eides statt, dass ich diese Dissertation selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Datum Muhammad Amir Mehmood

Abstract

The ubiquity of high speed Internet access, proliferation in the adoption of mobile devices, and the popularity of content-rich applications have brought a new dimension to the Internet landscape. Indeed, high speed residential connectivity and mobile wireless access have changed user expectations. The broadband access technologies (i. e., DSL, WiFi, 3/4G, and LTE), and smart mobile devices (i. e., Android, iPhone, iPad, etc.), have enabled users to interactively browse, stream videos (i. e., YouTube and Netflix), play online games, and share content for social networking.

All trends together have caused a fundamental change in how users interact with the Internet. Any adverse impact due to high Internet traffic, heterogeneous access, and application protocol mix on flows of different applications can result in sub par network performance and unsatisfactory user experience. Understanding the relation between network performance and user perception is thus crucial for application designers, network, and service providers. In this thesis, we endeavor to explore the impact of emerging network effects on different applications both from network performance and user experience point of view.

Studies of network performance and user experience require a multi-purpose heterogeneous testbed that supports a variety of networking conditions commonly present in today's Internet. We propose the design and architecture of *QoE-Lab*. The main features of QoE-Lab include 1) Next Generation Mobile Networks (NGMN), i. e., WiFi and 3G UMTS, 2) access/backbone network emulation, and 3) virtualization. It provides services like traffic generation, topology emulation, and high-precision cross-layer monitoring. We describe two Quality of Experience (QoE) case studies to show the benefits of the QoE-Lab testbed framework.

Next, we perform a sensitivity study of the packet loss process within a router for different network load levels, flow size distributions, and buffer sizes. We compare the loss process for TCP and UDP flows at different link utilizations and buffer sizes. We highlight the importance of understanding the flow-level properties of the traffic, e.g., packet loss under different networking conditions and their consequences on application performance, i. e., *flow-happiness*. We find that packet losses do not affect all flows similarly. Depending upon the network load and the buffer sizes, some flows either suffer significantly more drops or significantly less drops than the average loss rate.

Based on anonymized packet level traces from more than 20,000 DSL lines, server logs from a large content distribution network (CDN), and publicly available backbone traces, we investigate the flow-level performance of popular applications across a range of size-based flow classes. We use retransmissions, throughput, and round-trip-times (RTTs) as key flow performance metrics. We compare these metrics under different network loads, DSL link capacities, and for up/downstream directions. We show that irrespective of the direction, flows are severely impacted by events related to network load and application behavior. We also find that, in general, this impact (as measured by our performance metrics) differs markedly across the different flow classes. In particular, contrary to popular belief, small

flows from all applications, which make up the majority of flows, experience significant retransmissions, while very large flows, although small in number, experience very limited retransmissions. In terms of application-related performance, we observe that especially when compared to HTTP, P2P flows suffer from continuously high retransmissions. As for the root cause of these retransmissions, we identify the access part of the network as the main culprit and not the network core.

Further, we focus on the impact of networking conditions due to the adoption of heterogeneous wireless access technologies such as WiFi and 3G UMTS. These technologies have different physical layer characteristics and impose different networking conditions on flows. We study the impact of network handovers, codec switchover, and packet loss on VoIP applications. We compare our subjective test results with the wideband Perceptual Evaluation of Speech Quality (PESQ) prediction model. PESQ is often used as a standard for new generations of smart phones for quality assurance. We find that the WB-PESQ model underestimates the auditory quality in certain NGMN conditions, e. g., for wideband to narrowband speech codec switching.

In addition, we explore the impact of access networks, network handovers, video codecs and codecs changeover, video bit-rate, and bit-rate switching. Our work highlights the bottlenecks in video delivery over NGMNs. We find that network handovers have mostly a negative impact on user perception even if the video transmission is not affected by packet loss. In addition, the choice of video codec influences the video quality. While H.264 provides higher overall quality in WiFi networks, MPEG-4 improves user experience in 3G UMTS. Moreover, changing the video codec during a lossless transmission generally degrades the user experience.

Finally, we study the impact of NGMN conditions on web streaming video. We aim to understand how different access networks influence transport protocol (TCP) metrics and impact web video streaming quality. We complement the QoE estimations with network Quality of Service (QoS) parameters such as throughput, delay, and transport layer statistics. Our results show that 1) video QoE remains stable in WiFi even with high packet loss, 2) QoE in 3G UMTS is sensitive to packet loss even for low loss rates due to high variations in the network QoS, namely, throughput and delay, 3) the decrease in QoE and QoS in 3G UMTS is due to its negative interactions with the aggressive congestion control of CUBIC TCP, and 4) handover from WiFi to 3G UMTS degrades QoE.

Zusammenfassung

Die Allgegenwärtigkeit von High-Speed Internetzugängen, hohe Zuwachsraten bei der Nutzung von Mobilgeräten sowie die Popularität von inhaltsreichen Anwendungen haben der Internetlandschaft neue Dimensionen hinzugefügt. Damit einhergehend haben sich die Erwartungen der Nutzer durch schnelle Heiminternetanschlüsse und den mobilen, drahtlosen Netzzugriff geändert. Die Breitbandzugangstechnologien (z.B.: DSL, WiFi,3/4G und LTE) in Kombination mit smarten Mobilgeräten (z.B.: Android, iPhone,iPad) ermöglichen es den Nutzern das Internet zu durchsuchen, Videos zu streamen (z.B.: YouTube und Netflix), Onlinespiele zu spielen und Inhalte über soziale Netzwerke zu verbreiten.

Zusammen führten diese Trends zu einer fundamentalen Änderung in der Art und Weise in der Nutzer mit dem Internet interagieren. Hoher Internetverkehr, Schwankungen der Zugriffsraten und Anwendungsprotokolle verschiedener Anwendungen sind mögliche Faktoren, die sich in unterdurchschnittlicher Netzwerkleistung und nicht zufriedenstellendem Nutzerleben (User Experience) niederschlagen können. Entsprechend ist das Verstehen der Zusammenhänge von Nutzerleben und Netzwerkleistung für Anwendungsentwickler, Netzwerk- und Servicedienstleister essentiell. Daher wird in dieser Arbeit der Einfluss aufkommender Netzwerkeffekte auf unterschiedliche Anwendungen aus beiden Perspektiven zu betrachtet: Der Sicht der Netzwerkleistung und der Sicht des Nutzerlebens.

Studien, die sowohl Netzwerkleistung als auch Nutzererleben einbeziehen, erfordern ein heterogenes Mehrzwecktestbed, welches eine Vielzahl der im Internet vorhanden Netzwerkbedingungen unterstützen kann. Zu diesem Zweck schlagen wir das *QoE-Lab* vor. Die Hauptmerkmale von QoE-Lab beinhalten 1) Next Generation Mobile Networks (NGMN) z.B.: WiFi und 3G UMTS, 2) Access/Backbone Netzwerk-Emulierung und Virtualisierung. Es bietet weiterhin Möglichkeiten wie Traffic Verkehrsgenerierung, Topologieerzeugung und Hochpräzisions-Mehrschicht- Überwachung. Wir beschreiben zwei Quality of Experience -Fallstudien (QoE, Qualität des Nutzerlebens), um die Vorteile des QoE-Lab Testbeds zu zeigen.

Anschließend führen wir eine Sensitivitätsanalyse durch, um den Paketverlust innerhalb eines Routers bei unterschiedlichen Netzwerkauslastungsraten, Flow Size-Verteilungen und Buffergrößen zu untersuchen. Wir vergleichen den Verlustvorgang für Transport Protokolle TCP und UDP bei unterschiedlichen Linkauslastungen und Buffergrößen. Wir heben die Bedeutung von Flow Level-Eigenschaften des Verkehrs hervor, wie z.B.: Paketverlust unter verschiedenen Netzwerkbedingungen und die Konsequenzen für die Anwendungsleistung, z.B.: *flow-happiness*. Wir berichten, dass Paketverluste nicht alle Flows gleichermaßen beeinflussen. Basierend auf Netzwerklast und Buffergrößen zeigen einige Flows entweder signifikant höhere oder signifikant niedrigere Paketdropraten im Vergleich zur durchschnittlichen Paketdroprate.

Auf der Basis von anonymisierten Paketdaten von mehr als 20.000 DSL Leitungen, Server Logs von den großen Content Distribution Networks (CDN) und öffentlich verfügbaren Daten von Backbone Netzen, untersuchen wir die Flow Level-Leistung von bekannten Anwen-

dungen über eine Vielzahl von größenbasierten Flowklassen. Als primäre Flowleistungsmetriken benutzen wir Wiederübertragungen, Durchsatz und round-trip-times (RTTs). Wir vergleichen diese Metriken für unterschiedliche Netzwerkauslastungsraten, DSL-Linkkapazitäten und Up/Downstream-Richtungen.

Wir zeigen das Flows, ungeachtet der Richtung, stark beeinflusst werden durch Ereignisse, die in Bezug stehen, zu Netzwerklast und Anwenderverhalten. Wir berichten weiterhin, dass sich dieser Einfluss (gemessen entsprechend unserer Metrik) in unterschiedlichen Flowklassen deutlich unterscheidet. Im Gegensatz zur weit verbreiteten Annahme müssen kleine Flows aller Anwendungen, welche den Hauptanteil aller Flows ausmachen, deutlich mehr Wiederübertragungen durchführen als große Flows. Eine kleine Anzahl von Flows führt zu einer deutlich reduzierten Anzahl an Neuübertragungen. In Bezug auf anwendungsspezifische Leistung beobachten wir das P2P-Flows unter durchgängig hohen Neuübertragungsraten leiden, insbesondere im Vergleich mit HTTP. Als den Hauptgrund hierfür identifizieren wir den Zugangsteil und nicht den Kernteil des Netzwerks.

Des Weiteren konzentrieren wir uns auf den Einfluss von Netzwerkbedingungen die in Zusammenhang mit der Einführung unterschiedlicher drahtloser Zugangstechnologien, wie WiFi und 3G UMTS stehen. Diese Technologien haben unterschiedliche physikalische Layer-Merkmale und legen den Flows unterschiedliche Netzwerkbedingungen auf. Wir untersuchen den Einfluss von Netzwerkübergang, Codecwechseln und Paketverlust auf VoIP-Anwendungen (Voice over Internet Protocol). Wir vergleichen unsere subjektiven Testergebnisse mit den Wideband Perceptual Evaluation of Speech Quality-Vorhersagemodell (WB-PESQ). PESQ wird oft als Standard benutzt um Qualitätszusagen für neue Smartphones treffen zu können. Wir beobachten, dass das WB-PESQ Modell die akustische Qualität in gewissen NGMN Bedingungen unterschätzt, z.B.: für Codecwechsel zwischen Breitband und Schmalband-Sprache.

Zusätzlich untersuchen wir den Einfluss von Zugriffsnetzwerken, Netzwerkübergang, Videocodecs und Codecwechseln, Videobitraten und Bitratenwechseln. Unsere Arbeit hebt die Engpässe bei der Videoübertragung über NGMNs hervor. Wir finden heraus, dass Netzwerkübergaben meist einen negativen Einfluss auf das Nutzerleben haben, obwohl die Videoübertragung keinen Paketverlust zeigt. Darüber hinaus beeinflusst die Wahl des Videocodecs die Videoqualität. Während H.264 in WiFi-Netzwerken eine insgesamt bessere Qualität bietet, verbessert MPEG-4 das Nutzerleben in 3G UMTS-Netzwerken. Das Wechseln des Codecs während einer verlustfreien Übertragung senkt weiterhin, die von Benutzer, wahrgenommene Qualität.

Schließlich untersuchen wir den Einfluss von NGMN Bedingungen auf Webvideostreaming. Wir analysieren wie verschiedene Zugangsnetze und Transport Protokoll-Metriken die Webvideostreamingqualität beeinflussen. Wir ergänzen QoE-Abschätzungen mit Quality of Service (QoS) -Parametern wie Durchsatzrate, Verzögerungsraten und Transport Layer- Statistiken. Unsere Ergebnisse zeigen dass 1) Video-QoE für WiFi-Netzwerke selbst bei hohem Paketverlust stabil bleibt 2) QoE bei 3G UMTS-Netzwerke aufgrund von hohen QoS Schwankungen, insbesondere hinsichtlich der Durchsatz- und Verzögerungsraten, selbst für niedrige Verlustraten sehr sensibel auf Paketverlust reagiert. 3) die Verringerung von QoE und QoS in 3G UMTS -Netzen in negativer Wechselwirkung mit dem aggressiven Staukontrolle von CUBIC TCP steht und 4) die Übergabe von WiFi nach 3G UMTS die QoE senkt.

Pre-published Papers

Parts of this thesis are based on the following peer-reviewed papers that have already been published or are currently under submission. I thank all of my co-authors for their valuable contributions. All co-authors have been acknowledged as scientific collaborators of this work.

International Conferences

MEHMOOD, A., SENGUL, C., SARRAR, N., AND FELDMANN, A. Understanding Cross-Layer Effects on Quality of Experience for Video over NGMN. In *Proceedings of IEEE International Conference on Communications (ICC)* (June 2011)

MEHMOOD, A., WUNDSAM, A., UHLIG, S., LEVIN, D., SARRAR, N., AND FELDMANN, A. QoE-Lab: Towards evaluating Quality of Experience for Future Internet Conditions. In *Proceedings of TRIDENTCOM* (April 2011)

MEHMOOD, A., LEWCIO, B., VIDALES, P., FELDMANN, A., AND MÖLLER, S. Understanding Signal-Based Speech Quality Prediction in Future Mobile Communications. In *Proceedings of IEEE International Conference on Communications (ICC)* (May 2010)

MEHMOOD, A., HOHLFELD, O., LEVIN, D., WUNDSAM, A., CIUCU, F., SCHNEIDER, F., FELDMANN, A., AND BRAUN, R.-P. The Routerlab – Emulating Internet Characteristics in a Room. In *Proceedings of 11th ITG Symposium on Photonic Networks (11. ITG-Fachtagung Photonische Netze)* (May 2010)

Workshops

WUNDSAM, A., MEHMOOD, A., FELDMANN, A., AND MAENNEL, O. Network Troubleshooting with Mirror VNets. In *Proceedings of IEEE Globecom 2010 Workshop of Network of the Future (FutureNet-III)* (December 2010)

LEWCIO, B., BELMUDEZ, B., MEHMOOD, A., WÄLTERMANN, M., AND MÖLLER, S. Video Quality in Next Generation Mobile Networks - Perception of Time-Varying Transmission. In *Proceedings of IEEE International Communications Quality and Reliability Workshop (CQR)* (May 2011)

Posters and Demos

LEVIN, D., WUNDSAM, A., MEHMOOD, A., AND FELDMANN, A. BERLIN: The Berlin Experimental Router Laboratory for Innovative Networking. In *Proceedings of TRIDENTCOM* (May 2010)

MEHMOOD, A., AGARWAL, S., SENGUL, C., AND FELDMANN, A. Mobile Video QoE in Future Mobile Communications. In *Proceedings of the Passive and Active Measurement Conference (PAM)* (April 2010)

WUNDSAM, A., MEHMOOD, A., FELDMANN, A., AND MAENNEL, O. Network Troubleshooting with Shadow VNets. In *SIGCOMM '09: Demo Session of the 2009 conference* *on Applications, technologies, architectures, and protocols for computer communications* (August 2009)

Technical Report

MEHMOOD, A., SARRAR, N., UHLIG, S., AND FELDMANN, A. How happy are your flows: an empirical study of packet losses in router buffers. Tech. Rep. 2012-07, ISSN:1436-9915, May 2012

Contents

1 Introduction

Studying the behavior and performance of Internet traffic has been a prime focus of the Internet research community for the last twenty years. The Internet, with its more than 2 billion users [12] and over 5 billion connected devices [11] manifests its importance in many aspects of our lives such as communication, business, entertainment, education and social networking. Overall, the growth rate of the Internet has been phenomenal. According to Ericsson – a leading mobile telecommunications equipment vendor, by 2020 there will be 50 billion connected devices [5] to the Internet, i. e., due to the emerging concept such as 'Internet of things'.

Among many other factors underlying the significant growth of the Internet [63], one prime factor is the users' urge to have integrated and rich communications through a single medium. Such user expectations are forcing media service providers to offer their services through the Internet as well their traditional means. However, supporting these services for the ever increasing number of users is challenging without innovations in the different areas such as infrastructure, high-speed wired and wireless access technologies, server deployment strategies and disruptive applications. Indeed, the last five years of development in the Internet has offered users diverse technologies for rich communications.

While the Internet has become very successful by embracing new innovations in various areas, its complexity has soared many times. The complexity, mainly stems from the lack of coordination among different players in the Internet while introducing a variety of new access technologies, devices, protocols, and applications. Consequently, with time there have been plenty of changes visible in the Internet: in the access network technologies, in the backbone technologies, in the way content is delivered, in deploying servers, global traffic management and in applications usage trends. By design, the Internet is flexible enough to accommodate new innovations in different areas but at the same time uncoordinated new developments pose many challenges for various other players involved in the Internet.

1.1 Changing face of the Internet

In order to grasp the different kinds of changes in the Internet and their potential impact, we now visit different areas of the Internet such as backbone infrastructure, access networks, protocols and applications.

First, we look at the backbone infrastructure of the Internet. The change in the backbone networks stems from the shift to broadband access technologies such as DSL, Cable, FTTH, WiFi, 3G/4G technologies from low-speed telephone modems to high-speed broadband [8, 9, 36, 172]. The rapid adoption of high-speed access technologies has especially stressed the Internet service providers (ISPs), necessitating the introduction of higher capacity backbone links, e. g., 100 Gbps [13] to cope with the explosive bandwidth requirements of the access technologies. The high capacity link upgrades ensure that backbone links are not the bottleneck. Likewise, to achieve such high capacity links, routers in the backbone have evolved to deliver hundreds of Terabits per second switching capacities [6]. However, the design of high capacity routers imposes memory constraints due to board space, limited I/O bandwidth and demanding power budget [2]. These constraints essentially reduce the router buffers of such high capacity routers 25 fold and can no longer support the memory requirements imposed by the bandwidth-delay product rule of thumb [104]. Although the links in the backbone are being upgraded to very high capacity, some of these links are not fully utilized. The reason for such under utilization is the operational engineering rule of not loading a backbone link more than 30-40 % to avoid delay and packet loss. The questions of the appropriate link loading and the buffer sizing in the routers is a concern for the backbone network providers offering service level agreements to their customers.

Also, the structure of the Internet has changed significantly in the past few years, mainly driven by the Internet economics. New trends in inter-AS traffic patterns and provider interconnection peering strategies emerged. Content providers and content distribution networks (CDNs) are directly peering with the consumer networks, bypassing the traditional transit network to save costs and serve users with lower latency. Similarly, several tier-2/3 ISPs are now excessively engaging in peering with each other either at various Internet exchange points (IXPs) or directly for cost effectiveness. The emergence of Hyper Giants such as Google, Microsoft, and Akamai is an evidence of the flattening structure of the Internet [117]. Even though, the links within the ISP networks are highly over-provisioned, the transit and peering links, i. e., “middle mile” where traffic is exchanged between different ISPs, CDNs and content providers can become a bottleneck especially in the busy hours [121]. Such bottlenecks are one of the potential causes that affect different applications of users whose packets traverse these congested links.

While the structure of the Internet and backbone networks are undergoing changes, advances in the access network technologies are enabling high-speed indoor and outdoor broadband Internet access. In recent years, access network technologies – DSL, Cable, WiFi, 3/4G, WiMAX, and Long Term Evolution (LTE) – are either already used or being rolled out in most parts of the developed world. For residential customers, different standards of Digital subscriber line (DSL) and Cable can deliver between 50-100Mb/s. Furthermore, technologies such as fiber-to-the-home (FTTH) can support even higher data rates. With new WiFi network standards, data rates up to 150 Mb/s per user can be supported. Similarly, LTE advanced is set to provide 1Gb/s peak data rate in 4G. At the same time, new portable devices with rich functionality are being offered to take advantage of high-speed mobile Internet access. Smart phones and tablets equipped with multiple sensors and network interfaces, e. g., iPhone, iPad, Android are just a few examples. With the sharp increase in the number of mobile hand held devices, a significant traffic share is coming from these devices [128]. It is anticipated that by 2015, the mobile data traffic will reach 6.3 Exabyte per month, a 26 fold increase from 0.24 Exabyte per month in 2010 [62]. These advances in the Internet underscore the need to develop an understanding of how heterogeneous access technologies are impacting users perception of different applications when used via mobile devices.

Besides technological advances, many diverse applications are playing a vital role in the Internet by enabling users to satisfy their communication needs. Alongside web browsing, such popular applications as Facebook, Google and YouTube are used for social networking, information lookup, content sharing and entertainment purposes. By providing free services, multimedia applications such as Skype, Apple and Google services are attracting a large user base. They enable users to easily setup VoIP and video calls over the Internet. Additionally, traditional entertainment means such as television, radio and movies are also increasingly available via the Internet. Indeed, it provides users with the ability to watch content at any time. Many service providers are now offering triple play services which consists of HDTV-based IPTV, broadband Internet services, and VoIP [36, 172, 176]. Another important factor regarding change in the Internet relates to the user choice of content. Traffic statistics from recent measurement studies indicate that a significant number of bytes contributed to the Internet traffic is due to the popularity of video content [117, 127]. The sudden rise of video as a popular content type in the Internet is primarily due to the availability of browser based video players using Adobe flash video format (FLV). Also many popular news and sports websites are using browser based web streaming for their news and sports feeds [146]. HTTP features such as “progressive download”, where video is played back without downloading the complete video, has made browser based video streaming prevalent in the Internet. Recent Internet traffic studies suggest that HTTP video may account for 25-40 % of all HTTP traffic [117]. As a result, multimedia services have changed the application mix in the Internet and have raised at the same time concerns regarding their performance. First, the large number of bytes contributed by video applications may cause congestion which can result in degraded quality for the users. In some cases, that may force operators to upgrade link capacities. Second, how does multimedia traffic interact with bursty Internet traffic.

As the Internet infrastructure and application are undergoing changes, transport layer protocols such as TCP are also adapted. Congestion control mechanism such as New Reno has been shown to under utilize the network capacity for fat and long pipes [189]. Thus, a number of modifications to the standard TCP have been proposed to adapt to the new conditions in the Internet. New congestion control mechanisms such as CUBIC [82] and Compound TCP [189] are proposed for high link utilization while ensuring fairness among flows and stability over fast and long distance links. Additionally, techniques such as selective acknowledgements (SACK) [50] and window scaling [105] have been added to TCP end points for efficient loss recovery and high throughput. The interactions between TCP and the network play a vital role in defining the end-user experience. Limited TCP receive windows or excessive timeouts can render a TCP connection useless. Therefore, it is crucial to understand how TCP reacts to different heterogeneous environments in the Internet.

Finally, with more heterogeneity in the network, packets in the Internet are traversing multiple technologies. For instance, data packets of typical users might be generated by a smart phone moving through different wireless access technologies or a laptop attached to high-speed DSL, traversing the wired backbone through many access and backbone routers, potentially through peering links and terminating at a virtual machine inside a data center. These particular examples underline a number of factors that can influence performance by introducing additional delays and/or packet losses.

All together these observations regarding the major changes in the Internet and trends in usage impact protocols and applications behavior in the wild. Consequently, end-users who are using different applications in the Internet can be affected due to the networking effects as a result of access technologies diversity, traffic properties, and small or too large router buffers. In summary, the heterogeneous combination of technologies creates an environment for data traffic whose “combined effects” on the end-user perception have not yet been studied.

Understanding the relationship between network performance and end-user experience is important to network designers and operators, application and protocol designers, and service providers. Due to the fact that end-users who based on their experience decide about the fate of any offered service, network operators and service providers are becoming more interested in studies that combine network performance and end-user experience. Assessment of user experience or user perception is commonly called Quality of Experience (QoE).

The necessity of understanding user QoE creates opportunities between two research areas – networking and quality engineering. However, such inter-disciplinary undertakings are challenging due to the different scientific methodologies adopted by these two research communities; for example, the HCI community likes to conduct tests with a small set of subjects and obtain the ratings of a particular service, whereas, the networking community is interested in finding metrics that explain performance by analyzing network traffic traces gathered from multiple vantage points that represents a large user population of the Internet [111].

The need to investigate the impact of networking effects on the performance of different applications, and their possible implications on the end-user experience motivates us to pursue an inter-disciplinary study. In this thesis, we provide both flavors by exploring user experience and network performance together. In general, this work aims to demonstrate that inter-disciplinary studies, such as ours, are essential for better understanding and to bridge the gap between the two different fields – networking and quality engineering.

1.2 Goals

The objective of this thesis is to address the impact of various networking conditions in the Internet on different applications. By studying in combination aspects related to network bottlenecks, application metrics, and end-user satisfaction level, we want to identify the factors critical for the performance of Internet services.

To develop a deep understanding of events that happen at various different layers during communication, as a first step we build a heterogeneous testbed framework. The salient design requirements of the testbed framework include emulating Internet conditions by generating realistic traffic with delays that are typically present in todays Internet. This testbed is able to provide precise and accurate monitoring, i. e., details from router buffer statistics, congestion control protocol events such as congestion window progression, and application level statistics i. e., peak signal to noise ratio (PSNR), jitter buffer statistics, etc. For the purpose of automated experimentations of a large number of runs and analysis we need a software tool-kit which manages these experiment runs and produces results for various types of analyses.

Furthermore, we explore user experience in the presence of different networking effects that are prevalent in the Next Generation Mobile Networks (NGMN) environment [143]. Therefore, this testbed is able to provide mechanisms such as network handovers between different access technologies during an ongoing VoIP or video session. While network handovers are inevitable due to user mobility, codec changeover can provide means to adjust quality of sessions according to the current networking conditions. Therefore, we need means to change codecs and bit-rates of applications. Another requirement for the testbed is to realize network conditions that cause network handovers, codec changeovers, bit-rate switchovers, and different packet loss rates with in one ongoing multimedia session.

Next, we address fundamental networking questions: i) What are the critical factors that impact flows of different applications, which can influence end-user experience. ii) To what extent different load intensities, buffer sizes, and traffic patterns impact flows, iii) Is there any difference between loss process for TCP and UDP flows under different scenarios?, and iv) Are packet losses equally distributed across all flows for different applications, links, directions, and DSL capacities? We explore these question by leveraging the strength of controlled testbed and real life traffic traces collected at three diverse vantage points that represent traffic of a large user population.

Finally, by employing subjective and objective approaches, we examine the impact of wireless access network heterogeneity, e. g., WiFi and 3G UMTS on user perception of the multimedia applications such as VoIP, video and web-streaming. Furthermore, we want to identify the challenges associated with the multimedia QoE of users in NGMN conditions. In particular, we want to explore the impact of vertical handovers between WiFi and 3G UMTS, and vice versa; codec changeover between low quality and high quality codecs; bit-rate switchover due to application; and the cause-and-effect relationship of these factors with cross-layer examinations.

1.3 Summary of contributions

1.3.1 QoE-Lab testbed framework

We first outline the design and architecture of *QoE-Lab*, a multi-purpose heterogeneous testbed that supports a variety of networking conditions to study network performance and user perception. QoE-Lab includes 1) NGMN networks, 2) access/backbone network emulation, and 3) virtualization. It provides services like traffic generation, topology emulation and high-precision cross-layer monitoring. We describe different open source software tools and special hardware such as the NetFPGA platform for designing such a testbed. The experiments are provisioned, orchestrated and analyzed by a tool called EXPAUTO which supports automated experimentation and analysis.

1.3.2 Happy flows in the Internet

Next, we investigate the impact of cross traffic on fundamental network properties such as buffer overflows by using our QoE-Lab testbed framework. Our work highlights the importance of understanding the flow-level properties of the traffic, e.g., packet loss under different networking conditions and their consequences on application performance, i. e., *flow-happiness*. We describe the loss process experienced by different classes of flows depending on their flow sizes. We find that packet losses do not affect all flows similarly. Depending upon the network load and the buffer sizes, some flows either suffer significantly more drops or significantly less drops than the average loss rate. Very few flows actually observe a loss rate similar to the average loss rate.

1.3.3 Flow performance in the wild

Based on anonymized packet level traces from more than 20,000 DSL lines, server logs from the large content distribution network (CDN), and publicly available backbone traces, we investigate the flow-level performance of popular applications across a range of size-based flow classes. We use retransmissions, throughput, and RTTs as key flow performance metrics. We compare these metrics under different network loads, DSL link capacities, and for up/downstream directions. We show that irrespective of the direction, flows are severely impacted by events related to network load and application behavior. We also find that, in general, this impact (as measured by our performance metrics) differs markedly across the different flow classes. In particular, contrary to popular belief, the small flows from all applications, which make up the majority of flows, experience significant retransmissions, while the very large flows, although small in number, experience very limited retransmissions. In terms of application-related performance, we observe that especially when compared to HTTP, P2P flows suffer from continuously high retransmissions. As for the root cause of these retransmissions, we identify the access part of the network as the main culprit and not the network core.

1.3.4 VoIP QoE prediction in NGMN

Further, we focus on the impact of networking conditions due to the adoption of heterogeneous wireless access technologies such as WiFi and 3G UMTS. Since, these technologies have different physical layer characteristics and they impose different networking conditions on flows, we thus study the impact of network handovers, codec switchover[1], and packet loss due to adverse wireless network conditions on the VoIP applications. Further, we compare our subjective test results with the PESQ quality prediction model. We find that the model underestimates the auditory quality in certain NGMN conditions: 1) wideband-narrowband speech codec switching, 2) speech signal fading during codec switching, and 3) talk-spurt internal time-shifting due to jitter buffer instability. By pointing out the impact of these degradations on the speech signal, we show the importance of potential improvements and adaptation of the wideband PESQ model for next generation mobile networks.

1.3.5 Video QoE in NGMN

Recent Internet traffic studies have shown the popularity of video content on mobile devices. Therefore, we focus on the impact of NGMN network effects on video quality. In this study, we explore the impact of access networks, network handovers, video codecs and codecs changeover, video bit-rate and bit-rate switching. Our work highlights the bottlenecks in video delivery over NGMNs and propose perceptual guidelines for video delivery for mobile scenarios. We find that network handovers have mostly a negative impact on user perception even if the video transmission is not affected by packet loss. In addition, the choice of video codec influences the video quality. While H.264 provides higher overall quality in stable WiFi networks, MPEG-4 improves user experience in 3G UMTS. Moreover, changing the video codec during a lossless transmission generally degrades the user perception. Our findings allow us to provide guidelines for mobile video delivery in, e. g., mobile IPTV and mobile video conference scenarios.

[1]We use changeover and switchover terms interchangeably in this thesis.

1.3.6 Cross layer effects on video QoE in NGMN

Besides conventional mobile video delivery using UDP as transport layer protocol, web video streaming by employing TCP (e. g., flash video) has gained popularity among mobile users. Indeed, YouTube mobile reported more than 100 million video playbacks per day [26]. Therefore we next study the impact of NGMN conditions on web streaming video. This study aims to understand how different access networks influence TCP transport protocol metrics and the impact of the transport layer on web video streaming quality. We use *CUBIC* TCP as a transport protocol, which is the default TCP variant, e. g., in Android phones. We complement the QoE estimations with network Quality of Service (QoS) parameters such as throughput and delay, and transport layer statistics. Our results show that 1) video QoE remains stable in WiFi even with high packet loss, 2) QoE in 3G UMTS is sensitive to packet loss even for low loss rates due to high variations in the network QoS, namely, throughput and delay, 3) the decrease in QoE and QoS in 3G UMTS is due to its negative interactions with the aggressive congestion control of CUBIC TCP, and 4) handover from WiFi to HSDPA degrades QoE.

1.4 Structure of the thesis

In Chapter 2, we discuss the background work. Chapter 3 describes the design goals and implementation of the QoE-Lab testbed frame work. In Chapter 4, we explore the impact of packet losses on flows of different applications in emulated environment. We present flow performance of popular applications in the Internet in Chapter 5. Our first QoE study related to VoIP in NGMN conditions and its implications on wideband PESQ speech prediction model is described in Chapter 6. We continue in, Chapter 7, with video quality in NGMN conditions by exploring the effects of network handovers and codec switchovers. In Chapter 8, we examine the cross layer interactions for video quality for NGMN conditions. We conclude, in Chapter 9, with a summary of the contributions in this thesis and future work.

2 Background

In this chapter, we first review trends in Internet traffic and application mix as well as key Internet traffic properties in Section 2.1. We discuss popular fixed and wireless access technologies in Section 2.2. Next, we briefly outline tools for data analysis in Section 2.3. We briefly discuss speech and video codecs used by popular VoIP and video applications in Section 2.4. Finally, we describe concepts of QoS and QoE and methods for quality assessment in Section 2.5. We note that we discuss research related to our work in the related work section of each chapter separately.

2.1 Internet traffic

In this section, we first take a look at the growth of the Internet traffic and applications usage in fixed and mobile networks. Then, we review some of key properties associated with the Internet traffic.

2.1.1 Popular applications in fixed and mobile networks

The success of the Internet is visible by its tremendous increase in traffic over the last decade. Recent studies have shown that Internet traffic is growing at a phenomenal rate of 32-45% annually [63, 117]. This growth is due to the combination of factors such as proliferation of ever-increasing access speeds, introduction of smart mobile devices, new applications and services. The high speed access penetration has increased with 310M fixed broadband and 590M (almost twice) wireless broadband subscribers in June 2011 [17]. The high-speed access has opened up new choices for the users, for instance, the number of HD (high definition) TVs continues to rise. Recent consumer surveys have shown that 38% of the US households have one TV set which is connected to the Internet via gaming console, set-top box, other computing device, or direct network connection [120, 172]. Similarly, users are embracing latest releases of smart phones and tablets with HD displays, e. g., iPad. The traffic share due to mobile devices is increasing at an exponential rate. Recently, Ericsson has reported that mobile data traffic has grown 28% between 3rd and 4th quarter of 2011 with a total (uplink + downlink) of 580 PetaBytes/months [36].

Internet traffic volume is mainly contributed by few popular application protocols. One key example is world wide web (WWW) which mainly relies on HTTP protocol. Similarly, file sharing applications use P2P protocols for exchanging content between peers. Over time many studies have explored trends in the usage patterns of different application protocols. While the rise of P2P traffic a decade ago posed tremendous challenge with respect to traffic management for ISPs [161, 178], recent traffic studies have shown that P2P traffic has declined and HTTP traffic dominates the Internet traffic with over 60% share [127, 172, 175, 176]. One primary reason is the popularity of one-click hosting services such as Rapidshare [43, 127] and legal measures against illegal content sharing through P2P. The significant increase in use of streaming video is another reason for high HTTP traffic. For P2P applications, Bittorrent and eDonkey protocols contribute majority of the traffic.

Another key factor that has changed the dynamics of traffic for ISPs is the popularity of real-time entertainment (RTE). The recent study by Sandvine Inc. [172] has shown that RTE dominates with the share of 58% of traffic volume in fixed access in North America. Two top applications are Netflix (32.9%), and YouTube (13.8%). Surprisingly, 15.6% of the RTE by volume is consumed on smart phones or tablets via WiFi connectivity available in homes, whereas 27.8% of all YouTube traffic is watched on hand-held devices. For mobile traffic, RTE traffic accounts for 50.2%. However, the proportion of YouTube, and Netflix in the mobile world is quite different with 24.99% and 2.06%, respectively. Moreover, MPEG streaming and audio streaming account for 6.58% and 5%, respectively. The high consumption of video and audio content illuminates the importance of these applications for users when they are using their smart mobile devices.

2.1.2 Internet traffic properties

With the rise of the Internet traffic in 1990, researchers have characterize Internet traffic and compare it with the traditional telephony traffic. Several years of traffic studies have shown that Internet traffic deviates from the Poisson model [157] and instead shows self-similar behavior and long-range dependence leading to burstiness at a wide range of time scales (microseconds to hours) [122]. Willinger et al. [199] provide a plausible physical explanation for the existence of self-similarity, i. e., it is due to the superposition of large number of alternating ON/OFF sources such as packet trains due to TCP flows, where ON and OFF periods are heavy-tailed with infinite variance. Jiang et al. [110] examine the causes of burstiness in the Internet traffic at short time scales in the range of 100-1000 milliseconds. They found that TCP self-clocking and network queues are responsible for burstiness up to round trip time of the connection. They suggest that burstiness at time scale below RTT is possible to mitigate with TCP pacing.

Crovella and Bestavros [64] showed that self-similarity is also evident in the world-wide-web. By analyzing data from multiple web-servers, their findings suggest that self-similarity in the world-world-web is due to user behavior. They show that not only web-transfers are heavy-tailed in nature but also users' think time between web-transfers is also heavy-tailed.

Paxson [153] has comprehensively investigated packet loss characteristics in the wide area by using active probes of 100Kbytes. Findings from his study include prevalence of out-of-order packet delivery, variations in packet transit delays, and occurrences of congestion periods at different time scales. Bolot [52] characterized delays in the Internet by using UDP packet probes. He found that packets in the Internet experience rapidly fluctuating queueing delays over small time intervals and packet loss process is random.

The packet loss process in the Internet has been widely explored by researchers from different perspectives such as measurement, inference, empirical validation, new protocol development, and modelling [42, 60, 144, 145, 203].

2.2 Access technologies

In this section, we briefly outline prevalent heterogeneous access technologies. We first take a look at the development of access technologies in the fixed network domain. Then, we discuss popular wireless access technologies.

2.2.1 Fixed broadband technologies

A variety of access technologies are available for broadband communications in the fixed network domain. These include DSL, cable TV, and fiber to the home (FTTH). The main components of DSL consist of Digital Subscriber Line Access Multiplexers (DSLAMs) installed in service providers offices and DSL modems installed at customers premises. The supported line speed on these lines varies according to the distance of the wire. In case of DSL, for low speed connections twisted copper pair medium is used and for high speed connections combination of fiber and twisted pair cable is used. ADSL standards (ITU G.992.1-5) define line speeds up to 24Mbps [90]. For higher speeds, VDSL standards (ITU G.993.1) provide speeds up to 100Mbps.

Cable access networks are also popular in many parts of the world. They provide Internet access to group of people over a shared coaxial cable medium. These cables are terminated at Cable Modem Termination System (CMTS). Cable networks use Data Over Cable Service Interface Specification (DOCSIS) version 2.0 and 3.0. While DOCSIS 2.0 allows download rates up to 42.88 Mbps, DOCSIS 3.0 allows more than 100 Mbps rates [187].

Although DOCSIS standards allow higher rates, the actual speed depends on the ISP offerings. The third popular access technology which allows higher access speeds is FTTH. With the need to replace aging copper network with the new fiber networks, FTTH is also seeing a gradual roll out with up to 100Mbps offerings by large operators [3].

2.2.2 Wireless broadband technologies

High demand for data by mobile users has forced vendors to introduce wireless technologies with high data rates. The ubiquity of WiFi in home networks is one example. Different WiFi standards such as 802.11abg [28] allow users data rates up to 54Mbps. Newer specifications of WiFi such as 802.11n [29] using MIMO – Multiple Input Multiple output can increase data rates with four antennas to 288.8Mbps. WiFi access points are often connected to DSL routers which provides uplink for these short-range wireless access. While WiFi equipment is becoming an essential part in the home networking, its abundance is creating problems such as interference in high-density areas. The availability of WiFi access at public places is also increasing.

The evolution of cellular and wireless broadband technologies, e.g., GSM/GPRS, UMTS, WiMAX, and LTE (Long Term Evolution) has enabled users to connect to radio access networks while they are on move. 3G UMTS is by far the most deployed wireless broadband technology with over 1 billion mobile subscribers. The main technology used in 3G UMTS is High-speed Downlink Packet Access (HSDPA) [30].

HSDPA provides a shared channel for transmission where users share downlink resources in the time domain. Higher data rates in HSDPA are achieved by using techniques such as higher-order modulation, rate control, channel-dependent scheduling, and hybrid automatic repeat request (HARQ) with soft combining [65]. Hybrid ARQ is a combination of forward error-correcting coding and ARQ. While HSDPA provides high speed in the downlink, the combination of HSDPA and enhanced uplink is known as HSPA (High speed Packet Access). Most common data rates available with 3G access are 7.2Mbps and 14Mbps for HSDPA and HSPA, respectively. Other major components that define the core of 3G/UMTS network architecture are Serving GPRS Support Node (SGSN), and Gateway GPRS Support Node (GGSN).

In 3G UMTS[1], data packets traverse over the air interface from mobile device to Node-B. Multiple Node-Bs are connected to Radio Network Controller (RNC) in the backhaul network which is responsible for managing resources such as capacity allocation, call set-up, switching, and routing. RNC is further connected with SGSN. Data packets are then passed from RNC to SGSN which is a core router for the access network through which mobile station is connected. SGSN handles mobility, logical link management, authentication, and charging functions. Finally, data packets are forwarded to GGSN which is a core router connected to the Internet via a firewall.

[1] We use Universal Mobile Telecommunications System – UMTS and High Speed Downlink Packet Access – HSDPA interchangeably.

```
12713364.9 0.175 X.9.211.97 Y.149.220.77 http 3414 80
tcp 1632 2665 X SF hADadf 14200 10 28400 20 111.1
```

Figure 2.1: Bro sample output with anonymized IPs

Likewise, 4G access technologies such as Long Term Evolution - LTE and LTE-advanced promise data rates of 100Mbps and 500Mbps, respectively. An in depth description of the evolution of 3G and LTE along with specifications can be found, e. g., in a book by Dahlman et al. [65].

2.3 Data analysis tools

When performing Internet measurements studies one challenging aspect is how to analyse large amounts of data. Analysis of big data sets requires a careful selection of tools that can properly function at scale. In this section, we outline details of some of the common tools used for network data analysis.

2.3.1 Bro

Bro [155], primarily developed by Vern Paxson, is an open source real-time network analysis framework designed for Network Intrusion Detection. It also provides comprehensive network traffic analysis capabilities and is able to handle large data set with over 20,000 lines of policy scripts. Bro is state full and can track extensive application network state. It has a policy-neutral core which can accommodate a range of detection approaches. With extensive logging events it provides forensics capabilities for network traffic. For example, it provides detailed level connection summaries for different application protocols.

Figure 2.1 shows an example output of statistics reported by Bro for a single connection. It includes start time, duration, originator IP and destination IP, originator port, destination port, application protocol, direction, TCP state, additional flags (e.g., to indicate payload data in both directions), payload bytes, packet counts in both directions, as well as a round-trip-time (RTT) sample. The state information captures the state of the hand-shakes. Both the initial three-way hand-shake as well as the final connection closing hand-shake. The RTT sample is an estimation of the round-trip-time as obtained from the initial TCP hand-shake.

Structural design of Bro as explained in [155] is shown in the Figure 2.2. Bro can be run on live network traffic or already captured trace files. This input is then processed by using libpcap which handles all the interactions with the operating system. It also provides means to apply different filters for packet capturing. Before running any analyzer, customized policy scripts are loaded with policy script interpreter. The main aim of a policy script is to define which events needs to be triggered while analyzing traffic.

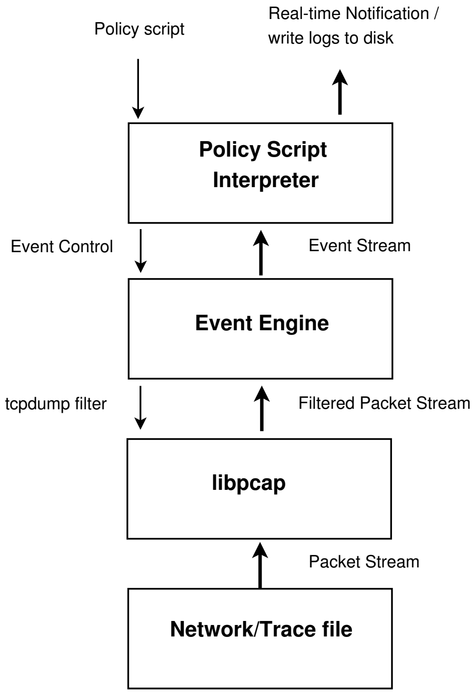


Figure 2.2: Design structure of Bro

The main functionality of the event engine is based on its powerful TCP reassembly and protocol analyzers. Every packet that is received by the event engine is parsed and associated with a connection. Multiple analyzers receive this packet for further processing. Analyzers in Bro detect application protocols by parsing the connections byte stream and matching it to multiple application signatures in general. Dynamic Packet Detection [67] is used for application protocol analysis. For each signature a specific analyzer is started which verifies that the bidirectional communication is consistent with the application layer protocol. Some of the protocol analyzers present in the Bro are ARP, UDP, TCP, ICMP, Bittorrent, DNS, FTP, Gnutella, HTTP, IRC, NFS, POP3, RPC, SMB, SMTP, SSH, SSL, etc. Complete description about bro features, policy scripts, and analyzers can be found at (www.bro-ids.org).

2.3.2 TCPDUMP

TCPDUMP is an open source command line tool used for analyzing packets from pcap files. It allows packet capturing and filtering based on different expressions. Using different command line options, it provides the description of the contents of packets. It can be used both for capturing packets from network interfaces and for analyzing packets from pcap files. The detailed description of Tcpdump features can be found at (www.tcpdump.org).

2.3.3 TShark

TShark [24] is another packet capturing and processing tool which mainly relies on Wireshark code base and libpcap. Besides capturing packets from the network it also provides different analyzers. It provides various types of statistics regarding packet streams for different protocols such as RTP, HTTP, DNS, etc. TShark is also used for debugging purposes.

2.4 Speech and video codecs

Speech codecs used for VoIP are classified into two categories: i) narrowband, and ii) wideband codecs. Narrowband and wideband codecs include audio frequencies in the range of 300-3400 Hz and 50-7000 Hz, respectively. Commonly used narrowband speech codecs are G.711 (64kbps) [94], G.723.1 (5.3 or 6.3kbps) [95] , G.729 (8kbps) [96]. In addition, for mobile environments the Adaptive Multi-Rate (AMR) codec is used which provides source rates from 4.75 to 12.2kbps for narrowband (AMR-NB) and bit-rates from 6.6kbps to 23.05kbps for wideband (AMR-WB) [167].

Similarly, a variety of video codecs are available for efficient video delivery. Most common video codecs include MPEG-4 and H.264 [2] [129]. H.264 provides an excellent video compression and requires less bit-rate (1.5 to 2 times) for the same video quality as that of MPEG-4. H.264 standard defines 21 profiles with different video bit-rates that range from low bit-rate to HD bit-rates. Popular video streaming services in the Internet rely on H.264 codec. Examples are YouTube, Apple HTTP live streaming, Hulu, Adobe flash based web-browsers, etc.

2.5 QoS and QoE

In this section, we first discuss quality elements associated with Quality of service (QoS) and we also look at how these quality elements are used by service providers. We next describe quality features related to Quality of Experience (QoE) and different methods of quality assessment.

2.5.1 Quality of Service (QoS)

Before delving into the concepts of QoS and QoE, we discuss definition of quality from the literature. Jekosch [108] has defined Quality as "*the result of judgement of the perceived composition of an entity with respect to its desired composition*". Similarly, ITU-T [149, 167] has defined the Quality of Service as "*the collective effect of objective service performance which ultimately determines the degree of satisfaction of a user of the service*".

[2] Also known as MPEG-4 Part 10 or AVC (Advanced Video Coding).

Important quality elements are based on network parameters such as throughput, latency, jitter, packet error rate, packet out-of-sequence rate, and packet loss rate [159].

In the Internet, traffic is naturally bursty. This essentially means that packets in the Internet can be subjected to long delays due to queuing, and can get dropped due to router buffer overflows. To meet different requirements of network quality of service, the Internet service providers offer certain levels of guarantees in the form of Service Level Agreements (SLAs) to their customers [14, 16, 130, 183]. These SLAs cover guarantees of certain levels of availability of service, latency, jitter, and packet loss over an agreed upon period, e. g., a month. Network SLAs are realized with different class-of-service (CoS) functions which provide different traffic classes with varying QoS levels [4]. Similarly, application service providers, e. g., Skype, rely on low bit-rate (proprietary) codecs, voice activity detection, and forward error correction techniques to provide better QoS and robustness [78].

In this thesis, we primarily use QoS metrics such as throughput, latency, packet out-of-sequence rate, and packet loss rates. These metrics allow us to differentiate between QoS levels received by flows of various different applications.

2.5.2 Quality of Experience (QoE)

According to the ITU-T P.10/G.100 [27], QoE is defined as "*the overall acceptability of an application or service as perceived subjectively by the end-user*". This includes many factors such as end devices, network, service infrastructure, user expectations, and the environment in which the user is communicating [182]. In the context of human computer interaction (HCI), Hassenzahl [85] has defined user experience or user perception as "*a momentary feeling (good-bad) while interacting with a product or service*". A good user experience can be achieved by fulfilling human needs and expectations from the product or service. User experience is highly dependent on factors related to user such as user's motivation, mood, expectations and the context [86]. With regards to multimedia user experience factors like continuity, directness, noisiness, and conversation effectiveness play a key role [195].

In general, there are two broad approaches for quality assessment, namely, subjective and objective quality assessment. We next take a look at both approaches separately.

2.5.2.1 Subjective quality assessment

Subjective quality assessment approach relies on the ratings of test users (subjects). In subjective tests, different stimuli in the form of speech samples or video clips are presented to the subjects. These stimuli can be presented to the users in isolation (absolute rating) or in pairs. In paired comparison quality is measured with respect to the reference stimuli [167].

For absolute category ratings (ACR) tests, ITU-T Rec. P.800 [99] has defined a 5-point ACR quality scale which is also called as MOS-scale. Subjects are asked to judge the quality of stimuli according to the *MOS scale*, i. e., between 1 and 5, where 1 means bad quality and 5 represents excellent quality. Quality ratings obtained from the subjects are then averaged to get the *Mean Opinion Score* – (MOS).

Subjective quality assessment approach requires a balanced set of sufficient number of subjects that represent different level of expertise, age groups and gender. Furthermore, presentation of stimuli in the form of pre-recorded audio or video clips is randomized while presenting to the users. In addition, multiple voice samples from different talkers are used in case of listening tests. Similarly, a range of different video sequences are utilized to eliminate biases of a particular stimulus. MOS results are then further processed to gain insight about the collective users response to the service. Different quality scales and details pertaining to subjective tests can be found in, e. g., [99, 167].

2.5.2.2 Objective quality assessment

While subjective tests provide the most accurate user experience assessment, they are not scalable. Therefore, objective quality assessment methods have been developed. The primary goal of the objective quality assessment approach is to model user experience based on signal properties, i. e., signal-based or service parameters, i. e., parameter-based.

The working principle of the signal-based quality assessment models is based on the comparison of the degraded signal received after the transmission with the full or reduced reference signal. Examples of signal-based models in speech domain include Perceptual Evaluation of Speech Quality – PESQ model (ITU-T Rec. P.862) [100] for narrowband. PESQ also has a wideband extension known as WB-PESQ model [101]. We will discuss PESQ and WB-PESQ in detail later in Chapter 6.

Similarly, signal-based models for video quality prediction such as J.144 [97] and J.341 [98] have been developed by ITU. Peak-Signal-to-Noise-Ratio (PSNR) and Structural Similarity Index (SSIM) [197, 200] metrics are commonly used for signal-based video quality assessment.

One of the drawbacks of signal-based models is that they require full-reference or reduced reference signal for quality prediction. To mitigate the requirement of reference signal, parameter-based models are used. In VoIP area, E-model (ITU-T Rec. G.107) [92] is widely used for quality prediction. Based on the information related to codec, packet loss rate, and delay, it predicts *R-value* which is later mapped to MOS scale in the range of 1-5. Due to its simplistic nature, E-model is often used for the call quality monitoring [25] and as a capacity planning tool. Recently, to capture new transmission effects for NGMN, a modified E-model is proposed by [125].

In the same vein, the examples of parameter based video quality assessments models are ITU-T G.1070 [93] and TV-Model [168]. These models often provide quality assessment for both audio and video. In this thesis, we rely on TV-Model and PSNR for video quality assessment besides subjective evaluation.

3

QoE-Lab testbed framework

3.1 Overview

Over the last three decades, the tremendous increase in the size of the Internet has determined the development of increasingly complex network protocols and configuration tools for its management. Therefore, it becomes increasingly important to explore the effects of possible design choices for a Future Internet, and also to be able to predict the current Internet's behavior to radical changes. To this end, there exist many theoretical models, simulations, and testbed environments, each with particular goals and capabilities which depend on how much they simplify the Internet's complexity.

Testbed environments have proved invaluable for developing new ideas, benchmarking existing protocols, validating theoretical ideas, and for debugging purposes. Both researchers and industry have used testbeds extensively as they offer reproducible and controlled experimentation. The challenges in building testbeds are associated with the preliminary requirements of the testbed. One challenge is to select the right hardware and software tools to meet the varying requirements of a broad spectrum of scenarios.

As discussed before, multimedia applications have recently been challenging existing mobile access networks and are raising the bar for next generation mobile networks, both in terms of network traffic as well as in the expectations of end users. At the same time, the network and server landscape sees changes due to the advent of virtualization and split-architecture, e. g., OpenFlow [132]. Furthermore, traffic characteristics are changing due to trends in the application protocols mix.

In this heterogeneous environment, quantitative measurement and prediction of the user's Quality Of Experience (QoE) require testbeds capable of studying these effects combined as well as in isolation, in a controlled and reproducible manner. Thus, these requirements translates into the necessity of a testbed which is flexible enough to create different scenarios and at the same time capture user perception, within the confines of the available resources.

Our goal is to broaden the specific conditions used in today's QoE experimentation to understand the user perception requirements for the future Internet. To this end, we present an integrated testbed called "QoE-Lab" which provides the ability to evaluate scenarios for the future Internet by combining all these new networking entities under different traffic properties with high-precision monitoring at different layers. It exposes applications to complex real networking conditions to gain insight about the user experience. We use both subjective and objective quality assessment approaches for QoE estimations.

The main contribution of our work is a heterogeneous testbed that enables evaluation of scenarios and the correlation of user perceived QoE with the networking conditions. It enriches the modular BERLIN [123] testbed framework with support for mobile next-generation wireless networks. QoE-Lab adds several QoE-specific services to BERLIN, including multimedia streaming for VoIP, video and generation of controlled background traffic with Internet backbone/access properties. It also improves the monitoring and instrumentation capabilities by providing high precision monitoring points both at the network, TCP stack, and application level. Among the effects that can be studied are network handovers between different wireless access technologies, the impact of dynamic migrations and resource contention in virtualized scenarios. All these effects can be studied combined as well as in isolation, with repeatable controlled background traffic patterns.

To orchestrate these components and provide repeatable experimentation we developed a software suite, called EXPAUTO that handles the setup, orchestration, monitoring, and analysis of the experimental data. To the best of our knowledge, this is the first testbed to address the following diverse goals together for quality perception studies: (i) different background traffic properties which are typical of access and backbone networks, (ii) time-varying channel transmission characteristics which are typical conditions of NGMNs, and (iii) including virtualized networking components in the backbone and edge networks. We believe that studies conducted on this testbed will provide new insights into the design choices for mobility management as well as service adaptation according to the user experience for future Internet scenarios.

We structure the rest of the chapter as follows. In Section 3.2, we explain the key components of the testbed. We discuss the testbed services in Section 3.3. The experimentation control plane, EXPAUTO, which manages experiments is explained in Section 3.4. We present two QoE case studies using our testbed framework in Section 3.5. We discuss related work in Section 3.6 and summarize our work in Section 3.7.

3.2 QoE-Lab architecture

Designing a testbed with diverse requirements is a challenging task. One critical task is to select suitable components that provide the right level of control at the software and hardware level. The main QoE-Lab components are shown in Figure 3.1. We use commodity hardware and open-source software in order to ease the reusability of components developed in the community and to be able to contribute to it and share our experience.

QoE-Lab is built upon the modular testbed architecture BERLIN [123]. Its layered structure enables the flexibility and experiment life cycle management required for QoE experiments in order to understand the relationship between user perception and network performance in future Internet scenarios.

3.2.1 BERLIN experimental hardware

BERLIN's hardware includes 30+ commodity rack servers with 2-8 cores, routers from Cisco and Juniper, and switches from Cisco, HP, NEC, and Quanta. Special purpose NetFPGA cards are available at a subset of servers. BERLIN features a hybrid, customizable physical topology. One part of the testbed is organized in a *router-centric* fashion for experimentation with commercial routers and current routing protocols. The other part is *switch-centric*, with devices fully meshed onto a manageable switching fabric with 200+ ports. This part is mainly used for experiments related to the clean-slate approach which is an idea of redesigning the Internet from scratch.

3.2.2 Integration of heterogeneous wireless access

For our QoE studies (Chapter 6, 7, and 8), we add state of the art wireless access technologies having similar characteristics of production-grade NGMNs. The main components of the NGMN part of the testbed consist of the Mobile Node (MN), the Correspondent Node (CN) and the Mobile IPv4 Home Agent (HA). The MN acts as the VoIP/video client and the CN as the VoIP/video server. All communication between the CN and the MN is managed by the HA. MN has the ability to connect to WiFi, 3G UMTS/HSDPA at the same time and maintain calls while roaming between these technologies. We will discuss testbed details specific to each QoE-study later in Chapter 6, 7, and 8. Our testbed framework allows us to perform subjective quality tests for multimedia applications related to fixed and mobile network conditions with users in online as well as offline manner.

3.2.3 Labtool management system

The Labtool [123] software management system provides a foundation for the QoE-Lab, sandwitched between QoE-Lab and the foundation infrastructure hardware. It serves as the

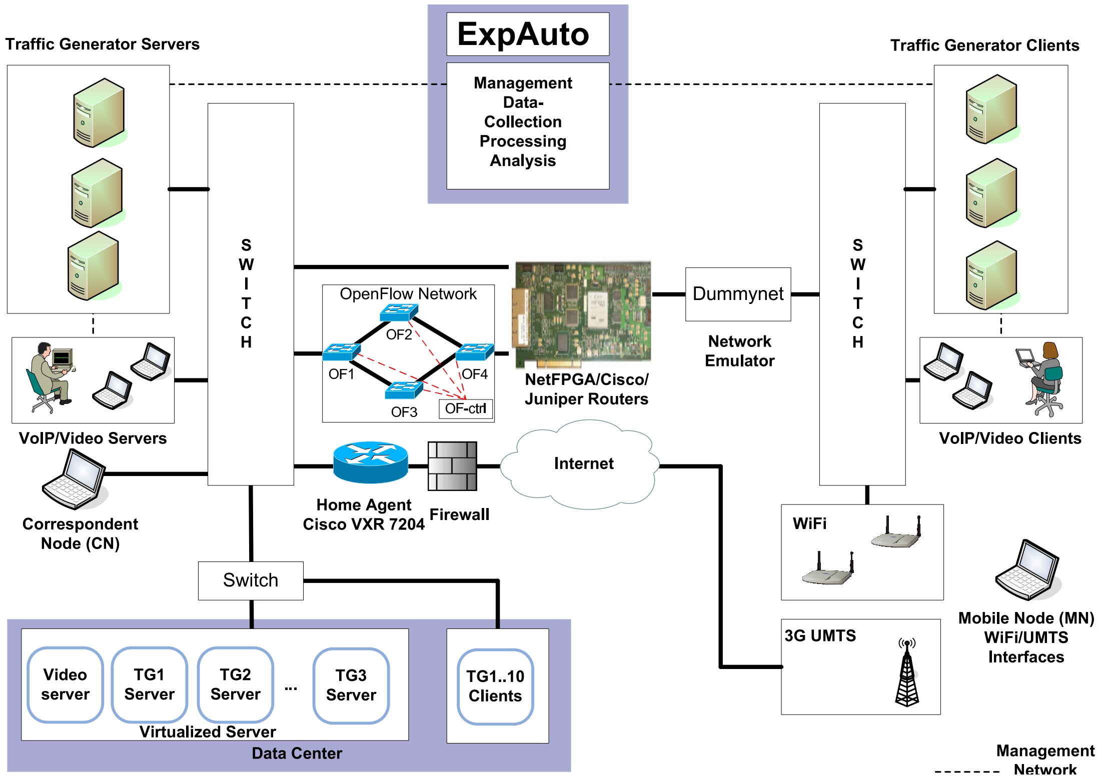


Figure 3.1: QoE-Lab overview.

primary interface through which users of the QoE-Lab reserve and simultaneously interact with their experimental hardware and perform system-level device configuration. The Labtool is experiment-centric, in that it organizes all of its functionality around the management, configuration, and repeatable deployment of experimental topologies and device configurations. The software architecture of the Labtool utilizes a three-layer client, server, and database structure, and is built to be extendable and scriptable with a client API in Ruby. Labtool provides the following functionalities:

Experiment life cycle management: The Labtool maintains an experiment-level view of all the actions it performs. This means that devices, the physical and virtual links connecting them, and their individual configurations are kept in the underlying database schema. This allows for easier hibernation, migration, and restoration of any particular experimental setup.

Physical topology versioning: The Labtool keeps track of all custom cabling changes over time and across experiments. Versioned cabling enables QoE-Lab administrators to alter and reliably restore topology changes.

Boot-mode configuration with imager system: An experiment performed on one set of devices should be repeatable on another set of suitably similar devices. To this end, the Labtool allows experimental device configurations for a given device to be redeployed onto any sufficiently similar device. The Labtool provides a collection of hardware-agnostic base operating system images which facilitate quick deployment of experimental environments.

3.2.4 Virtualized resources

Virtualization is one of key enabler that drives the evolution of data centers – the server side of the Internet. Thanks to initiatives like OpenFlow [132] it is also rapidly moving into the network domain, both in Enterprise networks and in ISP networks. This development brings many advantages for the operator, including more flexibility and agility in network management and potential for consolidation and reduced operational expenditures. However, these new possibilities can also create new challenges for the user experience, e.g., when flow setup times vary in OpenFlow, hosts are dynamically migrated, or traffic spikes in different virtual network influence each other due to insufficient isolation. To comprehensively study user experience scenarios in the evolving future Internet it is thus necessary to include a broad landscape of virtualization solutions. To precisely quantify the impact of Virtualization, a dedicated testbed is useful where cross-traffic patterns can be controlled and virtualized and non-virtualized approaches can be compared. Shared testbeds that rely on virtualization, e.g., Planetlab [147], are not ideal for such studies, as they do not allow precise control over the environment. QoE-Lab leverages the virtualization support built into BERLIN and provides solutions built onto XEN [46], VMWare ESXi [1] and KVM [116]. It also contains OpenFlow enabled hardware switches from three different vendors that can be flexibly integrated into the experiment setup.

3.3 Services

Based on the hardware architecture, QoE-Lab enriches BERLIN with important services for QoE studies, including controlled background traffic generation, topology emulation, and high-precision cross-layer monitoring. Next we discuss each service in detail.

3.3.1 Controlled generation of traffic with Internet characteristics

One of our main design objectives for the testbed is to study the impact of changing trends in traffic properties on user perception. We therefore need nodes which are capable of generating traffic which has similar characteristics to what can be observed in the Internet. To feed the network with background traffic, we rely on multiple PCs. We select Harpoon [184] as our network traffic generator because of its ability to reproduce flow-level behavior consistent with the Internet traffic characteristics. The two main parameters used for customizing Harpoon are the *flow-size distribution* and the connection *inter-arrival* time distribution. We note that TCP traffic makes up most of the traffic by bytes and most flows in the Internet rely on closed-loop feedback. We reflect this TCP traffic component in our traffic generation. One of the distinguishing features of Harpoon is its ability to use the underlying system's native TCP implementation, improving realism. While generating traffic which relies on TCP's feedback mechanism, it is also crucial to be able to change the *packet size distribution* for different experiments. At the time of execution, desired packets size distributions can be specified. Harpoon clients send Web requests for certain file sizes drawn from a pre-fed file size distribution to the Harpoon servers which subsequently responds by sending these objects. The superposition of these connections leads to the bursty behavior of traffic as seen in the Internet.

For traffic generation we use Intel Core2 Duo 2.20GHz servers with 2GB of RAM running Linux. Each server has two dual port Intel 82546 Gigabit Ethernet controllers. We use the 64-bit Linux kernel version 2.6.36. Each experimental machine has at least three network interfaces. One is exclusively used for controlling and managing the experiments while the other ones are used for traffic generation. Note that at the time of experiments we ensure no other traffic is present on the network. To create different network conditions we rely on different traffic load levels by changing the number of parallel Harpoon sessions. Note, increasing the offered load can lead to different link utilizations. To determine the necessary number of Harpoon sessions, we run the experiments without link capacity limitations. A Harpoon session is equivalent to flows generated by an Internet user. Our testbed allows us to generate more than 10Gbps of peak traffic load.

3.3.2 Topology emulation

The network topology we use for *backbone* network is the classical *dumbbell* one as shown in Figure 3.1. All network interfaces are one Gigabit Ethernet cards. The configurable network bottleneck is located between the NetFPGA router and the Dummynet network emulator. We use Dummynet [170] to add different round trip time (RTT) distributions and to configure different access bandwidth policies for emulating DSL clients with different access bandwidths. In addition, it is also used for creating conditions with packet loss impairments. We prefer Dummynet over NISTNet [59] for its better performance and reliability.

3.3.3 High-precision cross-layer monitoring

The reliability of experimental results highly depends on the accuracy and degree of the monitoring events during the experiments. Typical flow-level logging tools, such as *netflow*, do not provide precise enough information, e.g., about packet losses within a flow. Thus, we rely on custom high-precision monitoring in our testbed at various networking layers.

Router Buffers: Commercial router vendors such as Cisco and Juniper do not provide fine grained statistics about their router buffer occupancy. To circumvent this limitation, we use a NetFPGA [15] board as a router. A NetFPGA card is a special purpose network card with four one Gigabit Ethernet ports. As shown in Figure 3.1, two ports are used to interpose it between the Dummynet and the switch and a third port is used for VoIP or video traffic. Fourth port is used for connecting it to the host for capturing buffer statistics. It enables gathering highly accurate buffer statistics such as storing, removing and dropping packets at 8*ns* time granularity. In addition, buffer sizes and link capacities can also be controlled in NetFPGA devices.

Protocol Stack: We monitor the internal behavior of the TCP stack at Harpoon/Video servers using the *tcphook* [204] Linux kernel module, which is based on the In-kernel Protocol Sniffer (IPS). It provides a hook from user space into the kernel TCP implementation. All important TCP protocol stack status information, such as the TCP congestion window size, estimated round trip times and slow-start threshold – *ssthresh*, to be recorded. This set of information is vital for establishing cause and effect relationships between network performance and QoE.

Data capturing: We capture packet level traces at all the experimental machines. By comparing these captured traces, we are able to pinpoint missing packets along with transport layer information, e.g., TCP sequence numbers as well as timing information about when the drop occurred. In addition, we can observe all generated flows from the ingress and egress ports traces. This data enables us to understand per-flow loss process. Thus we can study the per-flow loss process.

Application: Going a step further, we also provide a monitoring for the jitter buffer within VoIP application. This view enables us to study packets that are received but affected by jitter and therefore not usable by the application. These effects cannot be observed by monitoring the networking layer only.In addition, for objective quality measures, we compute metrics like per frame Peak Signal to Noise Ratio (PSNR) for video. We also record the VoIP speech signals for validation of speech quality prediction models for NGMN conditions, such as network handovers, codec switchover and bit-rate switching.

3.4 Experimentation control plane

The experimentation control plane EXPAUTO of the QoE-Lab testbed is designed to run experiments in an automated manner for different networking conditions using the previously mentioned hardware components, and to analyze experimental data captured at different layers. It provides a unified user interface to run experiments parameterized by command line arguments, and is executed from a central computer with ssh access to all the experimental devices through a management network. EXPAUTO is implemented as a collection of bash scripts, whereas the analyzer consists of perl, python, and awk components. During each experimental run, details obtained, for instance via *sysctl* such as interface counters are recorded for experiments profiling. An overview of EXPAUTO is given by Figure 3.2. Its main features are: (i) configuration for experimental scenarios, (ii) measurements and trace collections, and (iii) automated analysis. We next describe each functionality in detail.

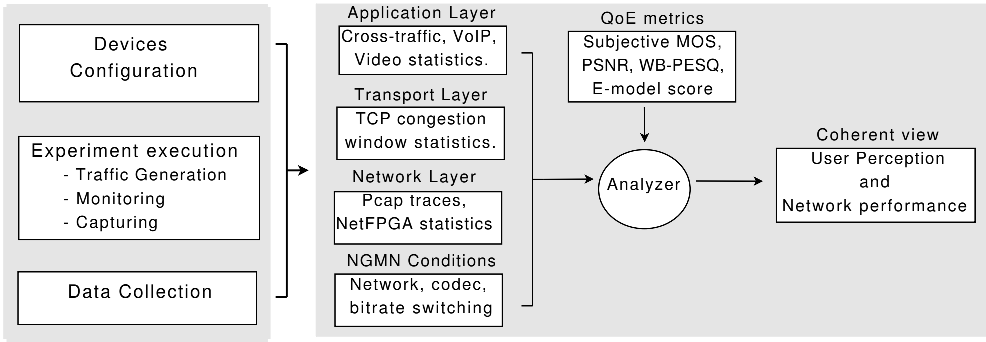


Figure 3.2: EXPAUTO architecture.

Configuration for experimental scenarios: To be able to create different combinations of networking scenarios with traffic load and burstiness, e.g., client-server where download traffic exceeds upload traffic or P2P where upload traffic is comparable to download traffic, we have pre-configured such scenarios in EXPAUTO. This means that the number of Harpoon processes are adjusted accordingly in both directions to match the traffic load. After receiving parameters through the command line interface, relevant configuration files for the selected scenarios are deployed on the experimental devices for traffic generation and network emulation. Dummynet nodes are configured to provide specific round-trip time distributions and packet delays. A separate management network is used for this control information. A separate process is used for controlling the VoIP/video application parameters.

Measurements and trace collections: After starting *tcpdump*, *tcphook*, and NetFPGA capturing processes on the experimental devices according to the experiments requirements, the experiments are executed for the desired duration. Once the experiment runs are finished, all processes are stopped and locally collected experimental data is transferred to a data storage server for further storage and post-processing.

Automation and Analysis: EXPAUTO orchestrates and collects data from several high-precision monitoring points in the devices across different layers, as described in Section 3.3.3. To obtain a consistent view of the experiment across multiple layers, it merges different network statistics from the NetFPGA buffer and packet-level traces. These statistics allow us to precisely reconstruct the behavior of individual flows at the transport-layer and correlate it to events that take place within the testbed hardware, e.g., inside router buffers. After experiments, EXPAUTO's analyzer post-processes the raw data and provide graphs using Gnu-R. These graphs visualize traffic properties, such as throughput at different timescales, RTT measured distributions and congestion window distributions, etc. Due to its modular design new analyzers can easily be added.

3.5 QoE case studies

In this section, we present results from two case studies carried out on the QoE-Lab testbed. Our aim is to qualitatively assert that each of the testbed component has a noticeable impact on the Quality of Experience (QoE) as perceived by the user. We start by showing the impact of different load regimes on audio and video QoE. We then illustrate the impact of common virtualization events such as virtual machine *migration* and host *overload* on video QoE. As split-architecture approaches like OpenFlow are currently gaining importance, we evaluate the impact of a prototype OpenFlow setup on video QoE under load.

3.5.1 Methodology

To illustrate the usability of our QoE-Lab, we present how various different networking components and networking conditions impact the multimedia quality of experience. To this aim, we use two types of network traffic in our QoE-Lab. For background traffic, we choose the Harpoon traffic generator for generating web-like workloads. In our experiments, Harpoon is configured with flow sizes generated from Pareto distribution with $alpha = 1.2$ and $mean = 1500$ (bytes). The inter-connection time has exponential distribution with mean $\mu = 1$ s. These two parameters entail that the generated traffic is bursty in a manner similar to Internet traffic. For multimedia traffic we capture a 55 second sequence from a real IPTV stream with video in standard definition (SD) resolution, encoded in H.264 and audio encoded using the mp2 codec, the standard format for high-quality portable video today.

We process this IPTV video stream so that it can be easily replayed by using *tcpreplay* [22] from any server in the Internet. We capture the multimedia stream by using *tcpdump* [21] at the receiver and extract audio/video quality metrics by using T-V-Model [168] – a parametric based IPTV quality prediction model. The T-V-Model extracts information about which packets are lost and based on the importance of the packet, it reports audio/video quality, i.e., Mean Opinion Score (MOS) for every 16 seconds interval where MOS value of 4.5 is the maximum and 1.0 is the minimum, i.e., unacceptable quality. We subjectively validate the MOS results by playing the video stream at the receiver in *VLC*.

We also capture Harpoon TCP traffic at the sender and receiver along with the TCP congestion window statistics at the servers and buffer statistics from the NetFPGA router. This monitoring enables us to understand the interactions between different networking layers and to establish relations between network performance and QoE. Each experiment lasts for two minutes. After starting the background traffic, we wait for 60 seconds to let it stabilize. We start the video streaming after this initial period and report the results for the second minute during which background traffic and video traffic compete for the limited bandwidth. During all the experiments, the bottleneck link capacity and the buffer size at the NetFPGA router are set to 242Mbps and 128 packets respectively. To emulate WAN conditions we introduce a delay of 155ms to every ACK packet of the background traffic

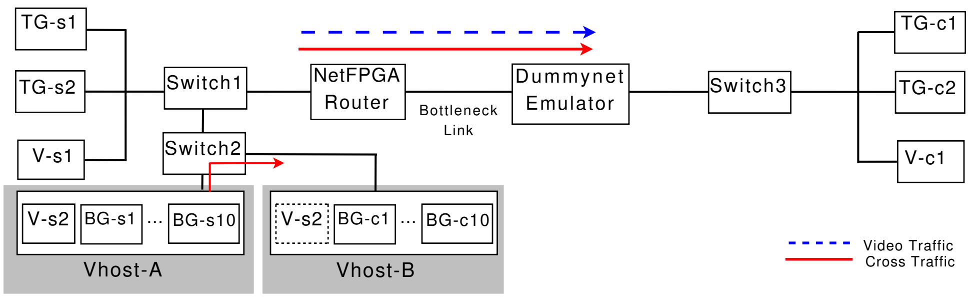


Figure 3.3: Setup for video QoE experiments with varying load and video server migration.

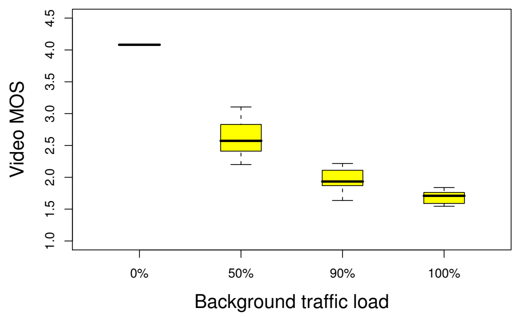
going in the direction from the clients to the servers. For every parameter setting, we repeat the experiment multiple times to get quality estimations.

3.5.2 Impact of background traffic on audio/video QoE

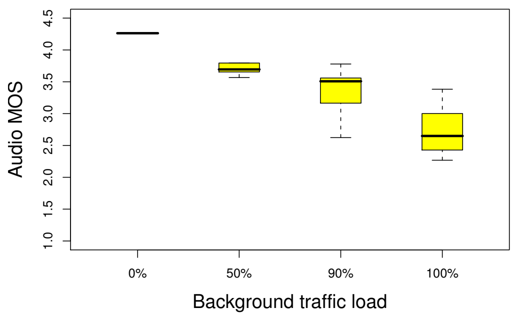
Here we aim to understand the impact of background traffic with different levels of burstiness on the Quality of Experience for audio and video applications. For this purpose, we use the servers TG-s1/2 and clients TG-c1/2 for background traffic generation as shown in Figure 3.3. Similarly we use video server V-s1 and video client V-c1 for multimedia traffic. During these experiments no traffic is generated from Vhost-A or Vhost-B. The results for audio and video QoE are shown in Figure 3.4. As a baseline, we first measure audio and video quality without background traffic, resulting in a MOS of 4.08. Then, we repeat the experiments by adding background traffic, leading to a link utilization of 50%, 90%, and 99% on the bottleneck link. It becomes apparent that video QoE is much more sensitive to network traffic load when compared to audio QoE. The main reason is that bursty packet losses within the video stream affect multiple frames resulting in poor visual quality, whereas in audio, losses are more easily concealed and recovered.

3.5.3 Impact of virtual server migration and overload

Service providers such as Google, Amazon and Microsoft all use large-scale data centers for delivering content to the clients, including on-demand videos. To optimize utilization, reduce costs and ease management, these data centers often consolidate several *Virtual Machines* (VMs) onto a single host that can be managed in a *cloud-computing* fashion. Some companies, e.g., Amazon, even rent out virtual machines to 3rd parties. For studying the impact of this virtualized environment, we use two virtualized servers Vhost-A and Vhost-B running XEN 3.3 with Ubuntu Linux 8.04 (2.6.24) as a Dom0. One virtual machine V-s2 runs on Vhost-A as shown in Figure 3.3. We now measure MOS results for different scenarios and present the results in Figure 3.6(a).



(a) Video Results



(b) Audio Results

Figure 3.4: Impact of background traffic on video and audio QoE.

In a multi-tenant setup where different entities share a single physical host, *performance isolation* between the different VMs is a critical factor. We investigate how the QoE provided by an on-demand video VM is impacted by background traffic on collocated VMs. We thus overload the active virtualization host (Vhost-A) by starting 10 additional virtual machines BG-s1 to BG-s10 and another 10 virtual machines BG-c1 to BG-c10 on second virtualization host (Vhost-B). One process of harpoon is started with 500 sessions on each BG VM to completely overload the link connecting Vhost-A and switch2. While video transmission was going on from V-s2 on Vhost-A to V-c1, 910Mb/s of background traffic from Vhost-A to Vhost-B was sharing the link in parallel. Even in this heavily overloaded setup, the QoE remains good with a median video MOS value of 3.8 shown as overload (OL) in Figure 3.6(a). This indicates the queue scheduling of XEN achieves good fairness for outgoing traffic. Complementary experiments show that fairness for *incoming* traffic can be much more problematic.

Virtualization also provides new management primitives that can affect QoE. VMs can be *provisioned* on-demand and after some time, once their utility is no more required, they can be destroyed to free the resources on the host. Virtualization solutions also provide the ability to *migrate* running virtual machines between running hosts, for load balancing or maintenance purposes. This migrations can either be performed in *offline* fashion, i.e., by freezing the VM on the source host, transferring its state to the target host, and then restoring it on the target host, or as *online* migration that minimizes downtime. In this mode, the source VM state is repeatedly and incrementally copied to the target host by the Hypervisor, while the VM keeps on running. When the source VM is finally frozen by the Hypervisor, only a small delta has to be transferred. We investigate how an ongoing migration of a VM acting as a video-on-demand server affects the QoE of the clients, both in online and offline fashion. We again use our two virtualized servers, Vhost-A and B, and a video serving VM starting out at Vhost-A. At the start of the experiment, video is streamed from V-s2 on Vhost-A and during the video session, the virtual machine V-s2 is migrated from Vhost-A to Vhost-B (see Figure 3.3). The comparison of video quality results indicate that offline migration (OM) has more impact on video quality as compared to live migration

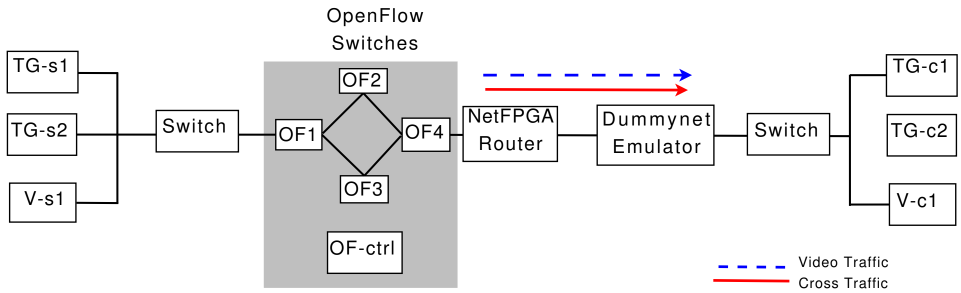


Figure 3.5: Setup for video QoE experiments with OpenFlow in the path.

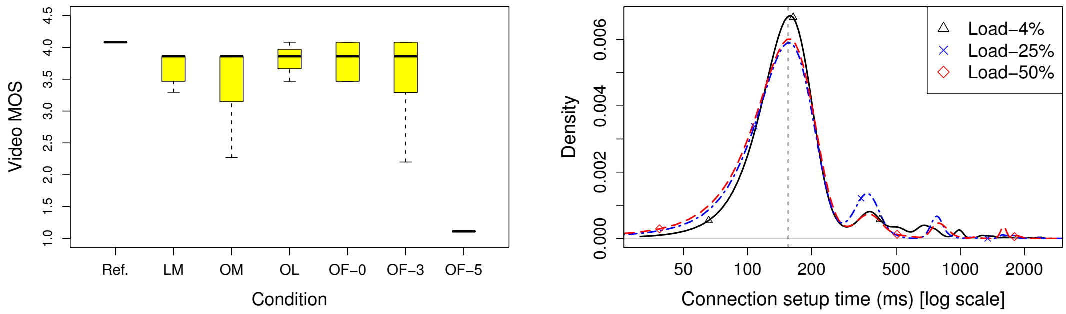
(LM) as shown in Figure 3.6(a). We observe that during migration, $1-2$ second of video is lost resulting in an abrupt video quality degradation.

3.5.4 Impact of a prototype OpenFlow setup on video quality

Recently, OpenFlow [132] has emerged as a disruptive approach to enable evolvable, software defined networks. OpenFlow is an open protocol that enables a commodity PC (the *controller*) to exercise flexible and dynamic control over the data traffic passing through Ethernet switches. To this end, the controller defines *Flow Rules* that specify the actions to be taken for matching packets. Flow rules can match on a 11-tuple of Layer-1 to Layer-4 properties, and can be installed *reactively* or *proactively*. Despite its prototype status, OpenFlow is already deployed in production networks in campuses and enterprises. It adds many degrees of freedom to the network management, particularly the *granularity* of flows and whether flows are installed in a *reactive* or *proactive* fashion. We investigate the potential impact of these decisions on QoE by using a simple reactive, fine-grained controller

As shown in Figure 3.5, we interpose a four node OpenFlow network consisting of prototype switches from two different vendors between the NetFPGA router and the switch. The switches are running OpenFlow 1.0 on prototype firmware, and have flow table sizes of around 2000 entries. They are managed by a dedicated NOX [81] controller running the `routing` module. This module implements shortest-path routing based on Layer 2 destination addresses, with a fine grained flow model, i.e., each individual TCP connection results in two flows. Flow rules are installed reactively. This is the common setup in use in many production OpenFlow networks today.

We are interested in the impact of this setup on the video quality in the presence of background traffic. We use the same methodology for background and video traffic as explained in Section 3.5.2. The results are shown in the right part of Figure 3.6(a). First, we send only one video flow without background traffic and we receive highest video quality MOS of 4.08 with little variation, marked as OF-0 in the Figure. Then we add background traffic of low average throughput, with 3Mbps (OF-3) and 5Mbps (OF-5). OF-3 achieves the same median MOS as OF-0, but shows more variation due to the burstiness of the background traffic. Surprisingly, for OF-5, the 2Mbps increase in background traffic results in a severe



(a) Virtualization networking conditions (b) Connection setup time for different load

Figure 3.6: Impact of virtualization on video QoE. LM:Live migration; OM:Offline migration; OL:Overload; OF-X, where X is background traffic load in Mbps

degradation of video quality, with a MOS of 1.1. This huge impact on video quality needs further investigation. It appears that in the case of an OpenFlow controlled network, quality is not a direct function of the network throughput as the load is significantly smaller as compared to the load in the experiments of Section 3.5.2.

We start our investigation by looking at the connection setup time, i.e., the time difference between SYN and SYN-ACK packets. As in this setup, the switch needs to contact the *controller* for a decision at every new flow arrival, connection setup time can be a critical factor. For our network we have introduced a RTT of 155ms as explained in Section 3.5.1. The flow setup time for traffic loads of 4%, 25% and 50% in case of OpenFlow is shown in Figure 3.6(b). At 25% load, the flow setup time has multiple modes beyond the configured RTT of 155ms. We further look into the flow arrival rate per $1s$ time bin for different traffic rates going through OpenFlow. The results are shown in Figure 3.7. This figure shows the bursty arrival of flows at the OpenFlow switches. We notice that the flow arrival rate surpasses 250 flows/s for OF-5 and OF-10. Further investigation shows drastic drops in QoE for flow rates beyond $250/s$. While the exact cause for this behavior is still under investigation, preliminary results indicate that such high flow rates can overload the switch CPU, and inhibit timely cleanup of expired flow entries from the flow tables. This causes the flow table to overflow and causes random evictions of the video flow, which explains the drastic deterioration in quality.

We conclude that with today's hardware, the frequently used approach of a purely reactive controller with a fine-grained *flow model* can be highly problematic for QoE when faced with realistic, bursty background traffic. Further investigation with different controller logic, reactiveness patterns and flow definitions is clearly necessary, as well as continued improvement of the prototype hardware.

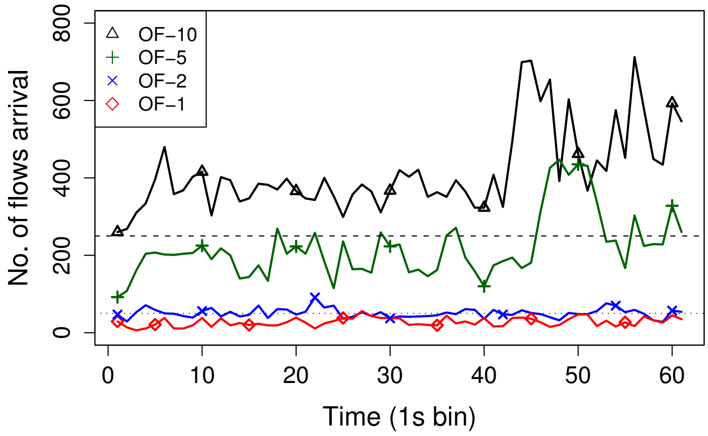


Figure 3.7: Flow arrival rate/sec for 1-10 Mbps traffic

3.6 Related work

Future Internet studies mostly rely on simulators, emulators and real testbeds. For QoE studies, testbeds are the most popular choices especially when subjects are involved. Our effort builds upon the previous effort of Mobisense testbed [192], which was primarily designed to assess the user perceived quality of mobility in NGMNs using VoIP. We extended this testbed to include QoE evaluation for both VoIP and video in the presence of controlled background traffic, virtualized resources and different NGMN conditions such as network handovers, codec changeovers and bitrate switchovers. WAIL [184] and Stanford testbed [47] were built to study the impact of traffic properties and router buffer sizes. However, the focus of these testbeds was not QoE evaluation.

3.7 Summary

In this chapter, we present the design and architecture of QoE-Lab, a modular testbed for the evaluation of the network performance and Quality of Experience (QoE) of different applications under the heterogeneous networking conditions. It features controlled background traffic generation, network emulation, seamless mobility between different access technologies, high precision monitoring, and an interconnection between experimental setup and the real Internet. It also includes hardware that is used in operational settings as well as emerging virtualization technologies, including OpenFlow. We also describe our software tool, EXPAUTO, which is used for creating different test scenarios, allocating resources, monitoring and collecting data at various networking layers.

Leveraging the strength of QoE-Lab framework, we present two QoE case studies. The first case study explores the impact of different load intensities on IPTV video quality. We find that video QoE is much more sensitive to network traffic load when compared to audio QoE. The main reason is that bursty packet losses within the video stream affect multiple frames resulting in poor visual quality, whereas in audio, losses are more easily concealed and recovered. Our second case study examines the impact of background traffic on emerging virtualization technologies and OpenFlow architectures. We find that offline migration of video servers has more impact on video quality as compared to live migration. In the case of an OpenFlow controlled network, quality is not a direct function of the network load rather it depends on the flow arrival rate. Furthermore, purely reactive controller with a fine-grained flow model can be highly problematic for QoE when faced with realistic, bursty background traffic.

4

Happy flows in the Internet

4.1 Overview

We now turn our attention towards evaluating flow performance in the Internet for various different scenarios. For this purpose, we decide to first explore several scenarios in a controlled testbed enviorment. Internet traffic has been shown to be bursty and, in particular, exhibits scaling properties, e.g., [152]. The possible causes include the ON/OFF process of the transmission times [122], the heavy-tailedness of flow size distributions [64, 74] , the multi-fractal behavior of TCP [110]. While, we, in this chapter, do not focus on scaling itself, we note that scaling appears to be an inherent invariant of realistic traffic. This has been repeatedly confirmed by thorough studies and empirical validations [53, 126].

Rather, we, in this chapter, focus on *packet losses*. Note that TCP makes packet losses inevitable. Standard variants of TCP use bandwidth probing to determine the available bandwidth on the network path. This implies that TCP increases its load on the network until it receives a signal, a lost packet, that the network can no longer support the increased load. Therefore, any simulation or experiments with TCP traffic with none or only very limited packet loss is suspicious since it implies that almost all flows are TCP window limited which is not the case in reality [127]. However, the loss process is not only impacted by the large time scale properties of the traffic process but also the small time scale effects, namely the ones before the knee in the scaling plots which are caused by RTT and TCP effects and are responsible for the multi-scale nature of traffic [74, 110].

So far, most studies of packet losses have focused on path losses [52, 144, 203, 206]. This chapter studies the loss process of a single network element. Our experiments show that the loss process exhibits scaling and leads to unfairness between flows. We show empirically that depending on the network conditions, e.g., small or large buffers, low or high congestion, the packet losses are not evenly distributed among the flows of different sizes. On the one hand large flows are positively discriminated. We call such flows *happy flows*[1]. On the other hand small flows are negatively discriminated. We call them *unhappy flows*.

Our methodology relies on tightly controlled experiments where we select specific congestion levels, flow size distributions, buffer sizes, and round-trip-times. Using advanced instrumentation via NetFPGA boards and careful analysis across network layers we can track the loss process and its impact on each individual flow and the flow's TCP congestion window state.

We use this setup to perform a sensitivity study of the effects of network load, flow size distribution, and buffer size on the traffic and note the following key insights:

Flow happiness: The losses observed by individual flows differ across flow sizes as well as within flow sizes, and depend on both the load and the buffer size. Moreover, any single flow is very unlikely to observe the global packet loss process.

Link utilization: Small buffers limit the size of the TCP congestion window, leading to poor link utilization.

Packet loss process: Packet losses are not simply random as assumed by stochastic models of TCP [42] but rather exhibit scaling effects under high load and are highly irregular under low load and large buffers. When buffers are small and the load is low, one can assume that losses are uncorrelated at time-scales below the typical round-trip-time.

The remainder of this chapter is structured as follows. We explain our experimental methodology in Section 4.2. We perform our global sensitivity study of load, buffer size and flow distributions in Section 4.3. In Section 4.4, we refine our sensitivity study on a per-flow basis and study the dynamics of the loss process. Related work is discussed in Section 4.6. We discuss some implications of our work in Section 4.5. We summarize our work in Section 4.7.

4.2 Methodology

Recreating the Internet like traffic characteristics with realistic hardware is a non-trivial task. Just by generating packets and overloading links do not represent the loss process as seen in the real Internet. In order to get Internet like traffic characteristics, we need to select several components in the testbed. These components include i) traffic generator that generates traffic with similar characteristics as seen in the Internet, e. g., based on TCP closed-loop, ii) topology that represents the Internet, iii) routers which can be instrumented

[1] We use the expression *happy flows* in the same vein as done for *happy packets* in [54].

to vary their buffer sizes and to gain access of their buffer statistics, iv) mechanisms of varying packet delays and, v) high precision monitoring at various layers of the protocol stack. Off-the-shelf routers do not provide details of buffer statistics or allow instrumentation due to proprietary reasons.

To achieve our goal of understanding the packet loss process, we again rely on our QoE-Lab testbed framework (Chapter 3). QoE-Lab is a configurable and flexible testbed that allows tightly controlled experiments along with precise monitoring on different layers of the protocol stack including router buffers. In this section, we describe QoE-Lab experimental setup used for studying packet loss process. We also present the design choices made for the hardware and software components.

The key components of QoE-Lab testbed framework related to packet loss study are as follows:

4.2.1 Topology emulation

Studies of network traffic often relies on network configuration where all clients are present on one side and all servers are present on the other side. The link between clients and servers then act as a bottleneck link. This arrangement ignores the presence of multiple links present in the path and confines to one bottleneck link. Such configuration is called a classical *dumbbell* topology. We also select the classical *dumbbell* topology as shown in Figure 4.1 for our experiments. All network interfaces are 1 Gigabit Ethernet cards. The configurable network bottleneck is located between the NetFPGA router and the Dummynet delay emulator. Harpoon clients sent Web requests to the Harpoon servers. Using Dummynet [170] we add a delay of 150*ms* to every ACK packet from the Harpoon clients to the Harpoon servers. This delay enables us to emulate round-trip-times which can occur in WAN environments [127]. We explicitly choose to focus on relatively large RTTs to better observe the impact of the buffer sizes and the delay imposed by TCP's feedback mechanism. Using the Bro network intrusion detection system [155], we examine the round-trip times within the TCP's three-way handshake and validate that the experimental RTTs have the expected distribution.

4.2.2 Realistic traffic generation

To feed the target buffer with enough traffic, we rely on multiple PCs. We selected Harpoon [184] because of its ability to reproduce flow-level behavior consistent with the Internet traffic characteristics. The two main parameters used for customizing Harpoon are the flow-size distribution and the flow inter-arrival time distribution. Most flows in the Internet rely on closed-loop feedback [163]. Therefore, we use TCP flows for most of the traffic. We also add some UDP flows using a VoIP client [18].

For traffic generation we use 4 Intel Core2 Duo 2.20GHz servers with 2GB of 667MHz DDR2 RAM. Each server has two dual port Intel 82546 Gigabit Ethernet controllers. We

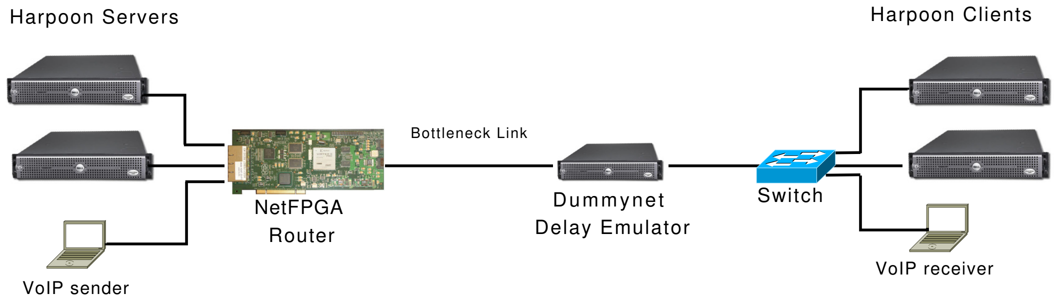


Figure 4.1: QoE-Lab experimental setup

use the 64-bit Linux kernel version 2.6.18 as distributed with Debian 4.0 (*Edgy etch*). Each experimental machine has at least two network interface cards. One is exclusively used for controlling and managing the experiments while the other ones are used for traffic generation. No other traffic was present on the network segments during the experiments. We use the default Ethernet MTU of 1500 bytes.

Harpoon is configured to choose file sizes according to Pareto distributions with shape parameters $\alpha = \{1.2, 1.5, 2.0\}$ and a mean of $\mu = 110KB$. These choices for the Pareto distribution ensure a finite mean while ensuring that the generated traffic exhibits variability and scaling behavior. To limit our parameter space, we choose to use an exponential distribution with mean $\mu = 1$ second for the inter-connection times, i.e., the user waiting times between different web requests.

To be able to compare losses seen by TCP with those from UDP, we generate UDP traffic with the open source VoIP framework *PJPROJECT 0.5.10.3* [18]. The generated speech files are using the $G.711$ codec.

4.2.3 Monitoring

Since commercial routers do not provide fine time scale statistics about their buffer occupancy we opt for the NetFPGA as a router. It allows to gather highly accurate buffer statistics. Moreover, we monitor the internal behavior of the TCP stack at Harpoon servers using the *tcphook* [204] Linux kernel module, for two reasons. First, if we want to study the impact of link congestion on the transport layer, we need the ability to monitor TCP's congestion window. Second, the initial value of the slow-start threshold tells us whether TCP is still in its first slow-start phase.

4.2.4 Network bottleneck

To ensure that the only bottleneck in our setup is the router buffer of the NetFPGA card, see Figure 5.1(b), we increase the maximum TCP receive window size to 20MB. This ensures

Table 4.1: Traffic generation parameters.

Load	Low	High	Very high
No. of Harpoon sessions	80	200	360
Offered load (%)	50	96	170
Average no. of concurrent TCP flows	140	1250	1700

that the transferred file sizes are not receiver window limited [47]. All experiments use TCP New Reno to control the size of the TCP congestion window.

4.2.5 Data capture

We capture packet level traces at both the ingress and egress ports of the NetFPGA router. By comparing both traces, we are able to pinpoint missing packets along with transport layer information, e.g., TCP sequence numbers, as well as timing information about when the drop occurred. In addition, we can observe all generated flows from the ingress port trace. Thus we can study the per-flow loss process. We run each experiment for 30 minutes. This duration allows each individual experiment to stabilize. The resulting traces, even though large, can be analyzed within a reasonable time.

4.2.6 Load

To create different network conditions we rely on three different load levels by changing the number of parallel Harpoon sessions on our clients. Note, increasing the offered load can lead to different link utilizations. We distinguish three load levels: *low*, *high*, and *very high*. To determine the necessary number of Harpoon sessions, we run the experiments without link capacity limitations. The lowest load, called *low load*, corresponds to a mean link utilization around 50% which should not impose too much congestion. However, once the load exceeds 50% one can expect degradations in the quality of service, e.g., increased delay and packet loss. Therefore we choose the *high load* scenario in such a way that the resulting utilization will be close to the link capacity. In the *very high load* scenario we intentionally overload the bottleneck link by letting the Harpoon servers generate about 1.7 times the capacity of the bottleneck link. The resulting number of Harpoon sessions[2] and the average number of concurrent TCP flows are shown in Table 4.1.

4.2.7 Buffer size

To help us choose which buffer sizes to rely on during our experiments, we take into account the recommendations provided by various buffering sizing studies. With our bottleneck ca-

[2] A Harpoon session is equivalent to flows generated by an Internet user.

Table 4.2: Buffer sizing recommendations for 1250 flows.

Buffer sizing scheme	BDP	Appenzeller	Tiny buffer
Buffer size (in packets)	3025	86	$20-50$

pacity of $242Mbps$ and round-trip time around $150ms$, the bandwidth delay product (BDP) suggests a buffer size of $3,025$ packets[3]. The scheme proposed by Appenzeller et al. [44] proposes the following equation to determine the buffer size:

$$B = \frac{RTT \times C}{\sqrt{N}} \tag{4.1}$$

where B is the buffer size in bits, RTT is the round-trip-time in seconds, C is the link capacity in bits per second, and N is the number of flows sharing the link. This leads to a buffer size of 86 packets for $N = 1250$ concurrent flows. The "tiny buffer" model [72] recommends buffer sizes around $20-50$ packets. For our experimentation, we mainly use buffer sizes of 256, 128, 64, 32 and 16 packets. We chose the upper bound of 256 packets for the buffer, as it is already 3 times as large as suggested by Appenzeller et al. [44]. During all our experiments, no packet loss occurs outside the bottleneck link. Table 4.2 presents buffer sizes calculated with BDP, Appenzerller's, and tiny buffer recommendations for 1250 flows.

4.3 Global sensitivity study

In this section, we describe the results of a global sensitivity analysis across our parameter space: traffic load, router buffer size, and flow size distribution. We start by studying the impact of different traffic loads and buffer sizes on the bottleneck link utilization (Section 4.3.1). Then, we show how much the buffer size impacts traffic variability and packet losses (Section 4.3.2). Finally, we point out the difficulty of sampling heavy-tailed flow sizes within reasonable length experiments for different traffic loads (Section 4.3.3).

4.3.1 Link utilization

We start our sensitivity study by examining how an exogenous variable such as link utilization is impacted by endogenous variables such as offered traffic load and buffer size. Note, that the link utilization is a consequence of both of these variables since in particular TCP is always trying to use all resources available in the network: the amount of traffic imposed by the flows that share the bottleneck link as well as amount of buffer available on the path between the traffic sources and sinks. Therefore, link utilization is a result rather than a directly tunable variable.

[3] The packet size used for the computations in this section is 1500 bytes.

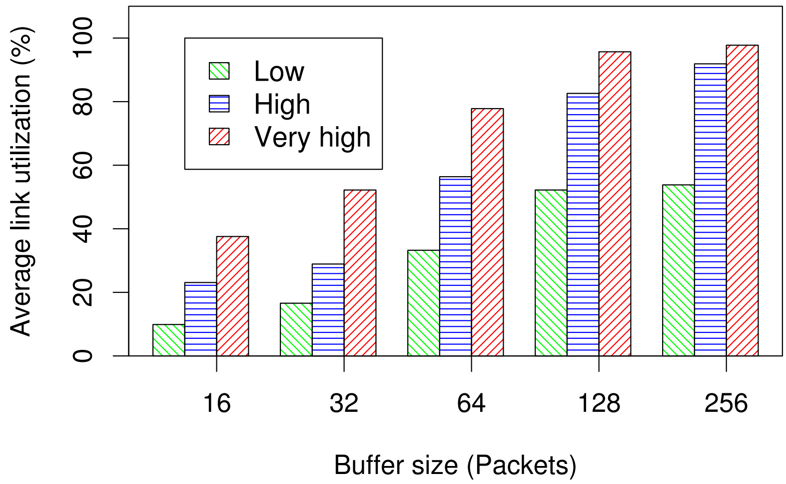


Figure 4.2: Average link utilization.

Figure 4.2 illustrates the impact of different router buffer sizes on the average link utilization for different offered load levels. Notice the impact of the buffer size. A limited buffer size prevents the traffic from utilizing the available link bandwidth even when many TCP flows are trying to push traffic across the network. Only the relatively large buffer sizes, i.e., 128 or 256 packets enable TCP to fully utilize the link capacity. When the router buffers are relatively small, i.e., 16 or 32 packets, TCP is unable to utilize the link capacity independent of the offered load.

However, the average link utilization does not tell the full story. We need to also examine the traffic variability. Thus, Figure 4.3 shows the corresponding standard deviations as well as the quantiles for Figure 4.2. The standard deviation is a first level summary of the variability and thus gives us a first indication of the impact of buffer size and load on traffic variability. With small buffers, TCP immediately suffers from losses when more than one flow is trying to send a burst of packets. Losses lead to reduced sending rates for each source and smaller overall congestion windows. Hence, the variations around the average link utilization are smaller. When router buffers are larger, each TCP source is able to send larger bursts without incurring losses, leading to higher variability in the traffic. When router buffers are large enough not to interfere too much with TCP's bandwidth probing mechanism, the increasing loads limit the possible variability when link utilization reaches the link capacity.

Figure 4.4 compares the link utilization across time for three different buffer size and load combinations at a time granularity of 1s. We chose these three combinations since they exhibit comparable average link utilization but correspond to different offered loads. The

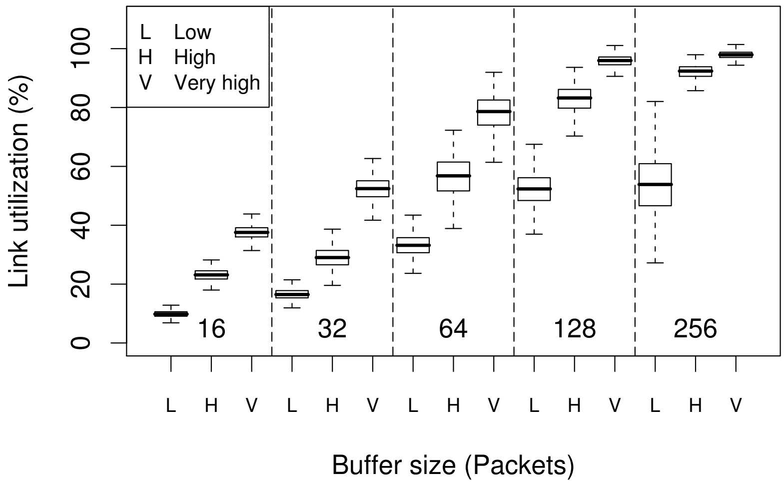


Figure 4.3: Variability in link utilization.

combination of a 256 packets buffer and low load exhibits very high variability. The combination of a 64 packets buffer and high load already has significantly less variability. Finally, the combination of a 16 packets buffer and very high load shows even less variability than the other two lines. This illustrates that the average link utilization alone is not sufficient to understand the traffic variability or the bottleneck.

Finally, we study the impact of offered load on variability. When the offered traffic load is high and router buffers are large, the link utilization is limited by the link capacity and thus traffic variability is also limited. Figure 4.5 shows the link utilization across time for 1s time bins for two experiments: one with low offered load and one with high offered load, for a buffer of 256 packets. For the high offered load, TCP is not limited by the buffer size. Instead, the link capacity does not allow the TCP senders to increase their rates.

4.3.2 Burstiness and packet losses

Internet traffic is known to be bursty at several time scales [64, 74, 110, 122]. Here, we are reexamining the traffic burstiness as seen by the router buffer to understand its impact on packet losses. When multiple TCP flows send bursts of packets at the same time, this can result in packet bursts that can exceed the router buffer size, leading to packet losses.

Micro-bursts: When examining the packet loss time series of the NetFPGA buffer we notice that large bursts of lost packets are often separated by a single packet that is suc-

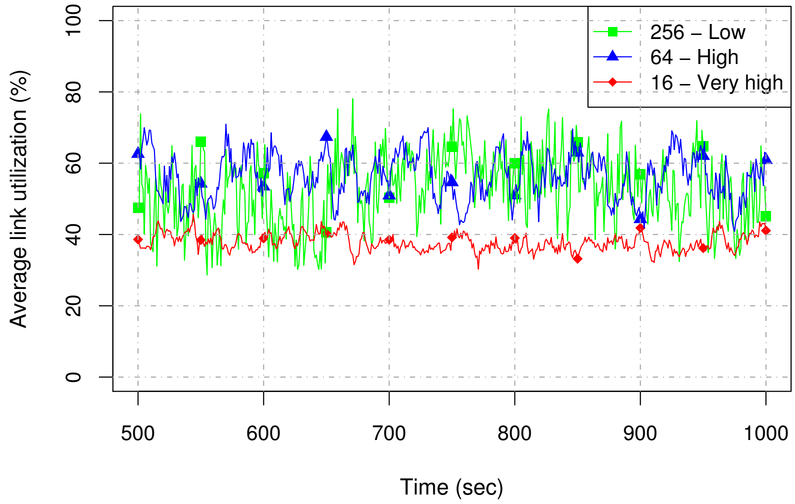


Figure 4.4: Traffic variability.

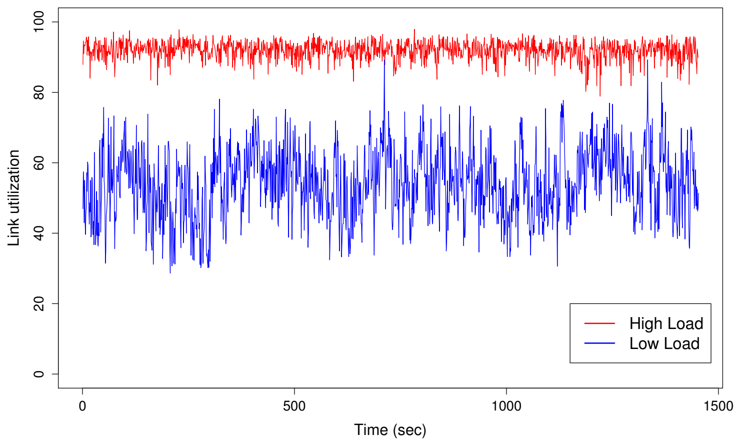
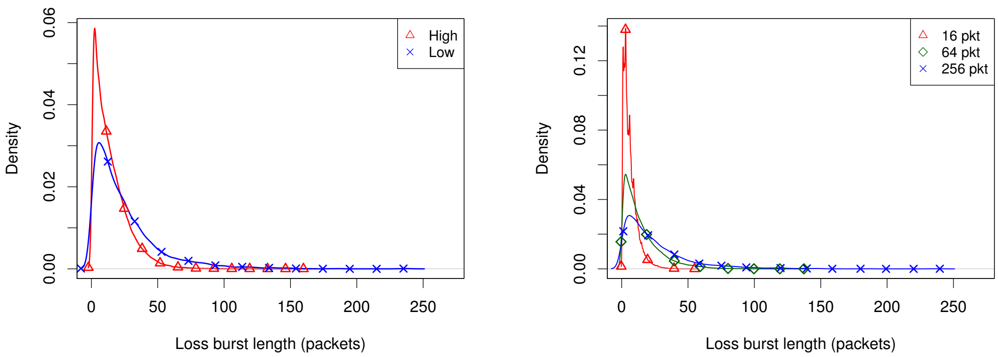


Figure 4.5: Impact of load on traffic variability.

cessfully sent by the NetFPGA card between two *micro-bursts*[4]. This phenomenon leads to an underestimation of the lost packet burst sizes. Moreover, it leads to loss burst length distributions that are unexpectedly multi-modal. Therefore, we stitch together *micro-bursts* of packet losses separated by a single successfully delivered packet. Using more than one

[4]Note, a micro-burst of lost packets can contain packets from a single or multiple parallel TCP flows.



(a) Different loads, 256 packets buffer (b) Different buffer sizes, low load

Figure 4.6: Distribution of loss burst length inside the router buffer.

packet during the stitching does not significantly change the loss burst length distribution. Under large buffers, e.g., 256 packets, the median micro-burst size is about 20 times smaller than the median of the stitched burst size under high load. Under low load, the ratio is about 36. As shown earlier, with large buffers and under low load, traffic is very bursty, as shown in Figure 4.3.

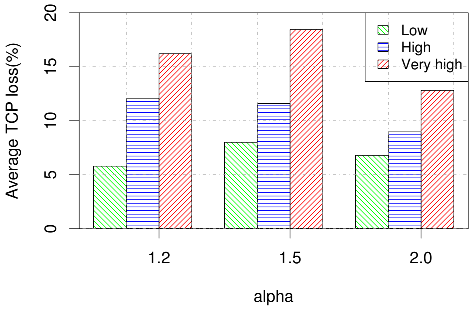
The resulting loss burst length distribution is shown in Figure 4.6(a) for a low and a high offered load scenario and a buffer size of 256 packets. We observe that under low offered load, the tail of the loss burst length distribution is heavier than under high load. This is expected given the higher ratios between the median micro-burst size and the stitched burst size under low load. Under low load, loss bursts are expected to be larger.

The distribution of loss burst lengths is affected by router buffer limitations in the same way as it is by high load. Figure 4.6(b) shows the loss burst length distribution for different buffer sizes and low load. Small buffer sizes lead to a similar distribution of loss burst length as under high load, with many more smaller loss bursts. Under small buffer sizes, TCP is limited by its congestion window.

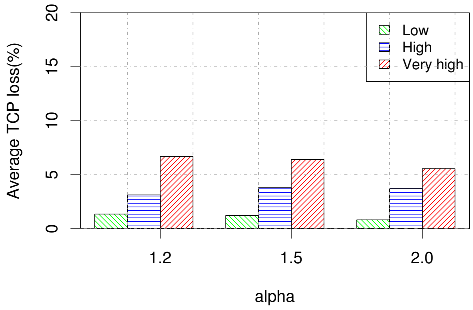
4.3.2.1 Packet loss

In principle, traffic burstiness is not a problem. However, since traffic burstiness leads to packet losses it might lead to substantially reduced performance for some flows. However, losses are inevitable with TCP. TCP estimates the available path capacity by generating losses and backing off once it detects a loss. We thus study the average packet loss under different buffer sizes, loads, and flow size distributions.

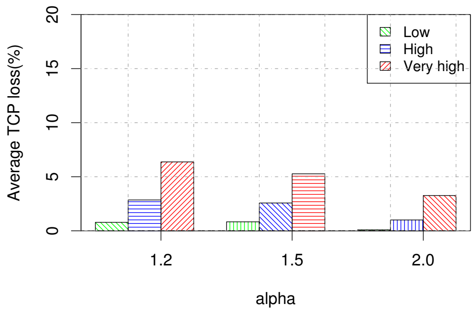
One of the contributors to Internet traffic variability is the heavy-tailed nature of flow size distributions [64, 74]. We therefore expect to see an impact of the degree of the heavy-tailedness of flow size distributions on the loss process. Figure 4.7 shows the average loss



(a) 32 packets buffer



(b) 128 packets buffer

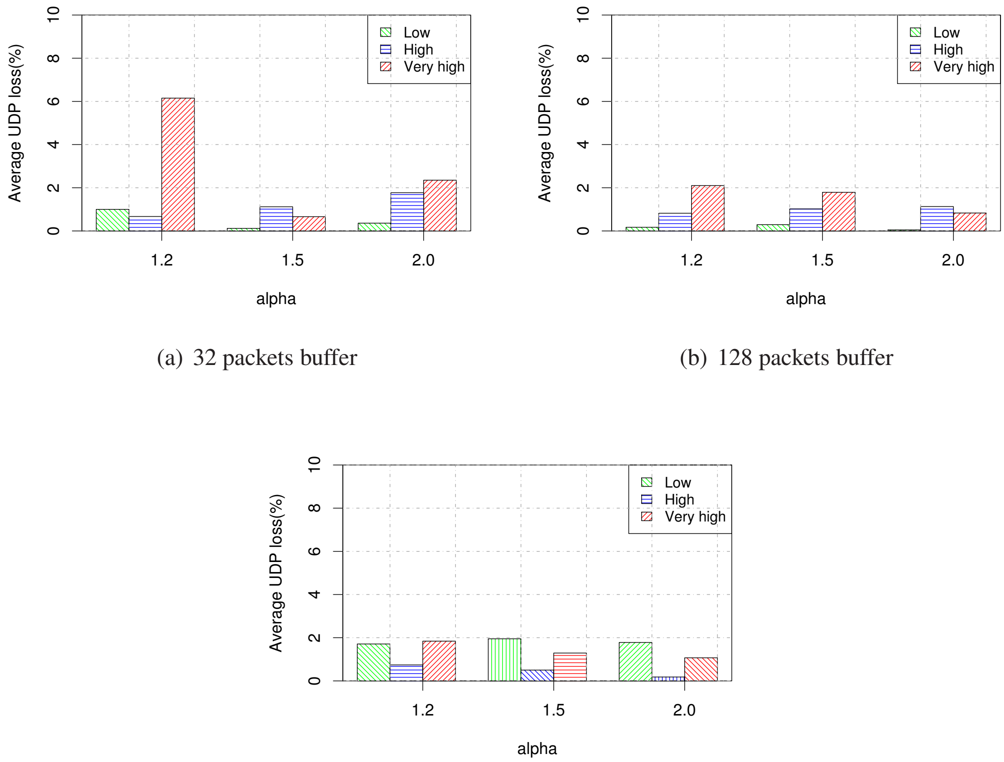


(c) 256 packets buffer

Figure 4.7: Packet loss observed by TCP.

observed by TCP flows for different buffer sizes, loads, and flow size distributions. Smaller buffers generate high packet losses, larger than 5%, even under low load. This happens because TCP is trying to estimate the available bandwidth on the link based on packet drops, while the drops occur not because of limited bandwidth on the link, but due to too small buffers that cannot handle the packets from the many concurrent TCP flows. When large enough buffers are available, the loss rate reduces dramatically, especially under low load (about 1% loss).

The impact of the heavy-tailedness of the flow size distribution is visible for large buffer sizes (128 and 256 packets). The lower the value of α, the heavier the tail of the flow size distribution, and the higher the packet losses due to a larger number of small flows and the few large flows. When traffic load is high or the buffer size is small the impact of heavy-tails on packet loss is limited by the way TCP is restricted in its burstiness. Note, larger buffer sizes show similar results to those experiments with 256 packet buffers.



(a) 32 packets buffer

(b) 128 packets buffer

(c) 256 packets buffer

Figure 4.8: Packet loss observed by UDP.

4.3.2.2 UDP and sampling the loss process

Most existing studies of packet loss in the Internet [52, 144, 203] rely on active measurements for sampling the loss process on Internet paths. Such measurements send packets at specific time intervals and infer the loss process based on the observed losses. Sampling the loss process in this way can suffer from two shortcomings: First, one may not sample the periods in which the buffer is full. Second, one may sample the periods in which there is a single free spot available in the buffer. To understand the impact of using such a sampling approach for estimating packet loss we generate UDP traffic using VoIP clients at a rate of about 250 packets per second. This rate is actually higher than used in the literature [52, 144, 203].

Figure 4.8 shows the average packet loss rate observed by UDP traffic. Note, the loss rate observed by UDP is in general much smaller than the one experienced by the TCP traffic. This is caused by two effects: first the fact that losses are usually occurring in bursts, see Figure 4.6 and the buffer occupancy process. However, when the router buffer is relatively large (256 packets) and the offered traffic load is low, UDP observes more losses than TCP. This effect is due to an unequal distribution of losses across flows of different sizes, which is examined in Section 4.4.

4.3.2.3 Buffer occupancy

The buffer occupancy gives additional evidence and an intuitive explanation for why different loss rates occur for different buffer sizes. For example, for low offered load and a relatively large buffers (Figure 4.9), we observe that the mode of the buffer occupancy is relatively small. This implies that for most packets entering the buffer, there is room. However, as the offered load increases, the mode of the buffer occupancy is shifted to the right. Therefore, each packet entering the buffer will have a non-negligible probability of being dropped. Similar results apply to larger buffer sizes as well.

4.3.3 Sampling heavy-tails

When traffic load is high or buffers are small, traffic variability is limited. Thus, large flows take more time to complete. If the duration of the experiments was infinite, this would not be a problem. In practice however, we have to limit the duration of our experiments. In this chapter, we chose to limit our experiments to 30 minutes, which allows us to sample most of the large flows while keeping the duration of the experiments reasonable and the traces manageable.

As mentioned in Section 4.2, we rely on different flow size distributions for our experiments: exponential and Pareto ($\alpha = 1.2, 1.5, 2$). When using heavy-tailed distributions such as Pareto, some flows are going to be very large. Indeed, some are so large that they may take longer than the duration of the experiment to complete.

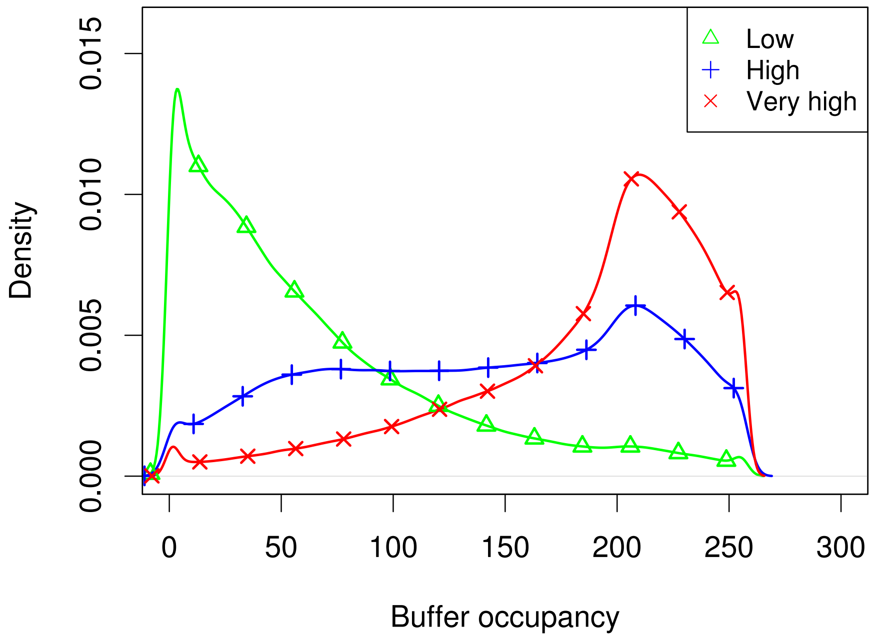
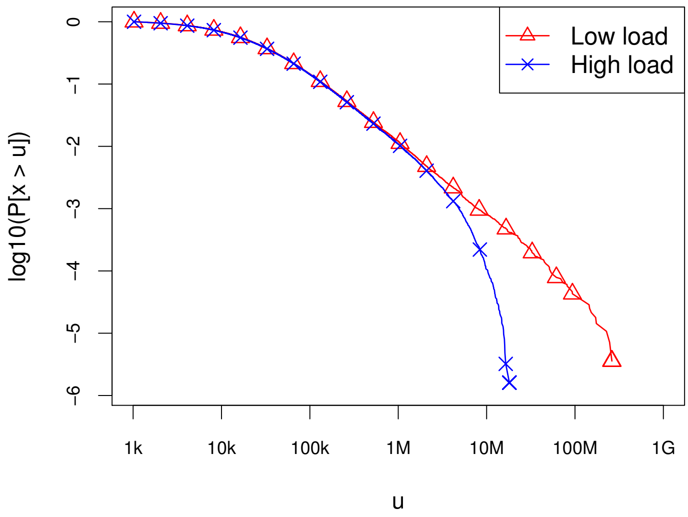


Figure 4.9: Buffer occupancy for different loads, 256 packets buffer.

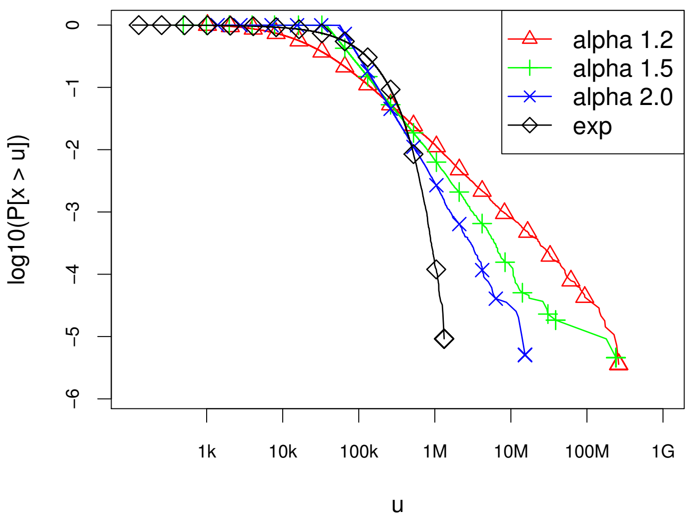
Figure 4.10(a) shows the impact of the offered load on the flow size distributions observed in the traces, for Pareto distributions with $\alpha = 1.2$. We observe that under low load, the tail is nicely sampled. Even flows as large as a few hundreds of MB complete. Under high offered load, flows larger than 10MB hardly complete. Fortunately, our sensitivity study indicates that the impact of the flow distribution on the packet loss process is limited. However, in general one has to pay attention to such sampling issues. Figure 4.10(b) shows the CCDF of flow sizes for the 4 chosen flow size distributions under low offered load and a buffer size of 256 packets. We observe that the tail is well sampled for all distributions, as expected.

4.4 Flow-level packet loss

So far, we have treated the loss process as a global phenomenon, i.e., one that takes places across all flows that share the buffer and one that does not change over time. However, we already observed in Section 4.3.2.2 that a limited rate packet flow that samples losses may see a different view of the loss process from one that has a global view of the router buffer. We study in this section the process of packet losses as it applies to each flow individually across different flows sizes and across time.



(a) Impact of load

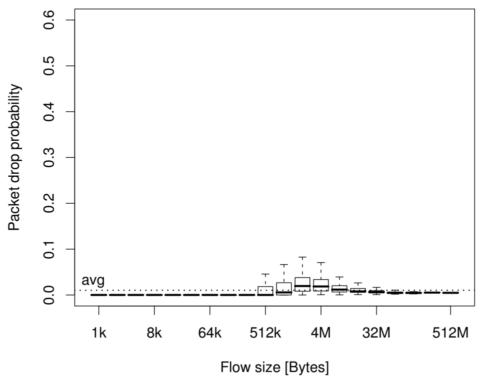


(b) Distributions under low load

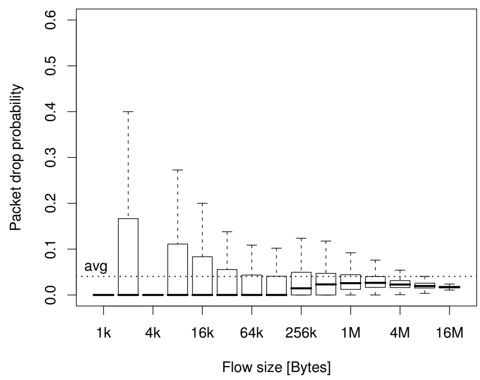
Figure 4.10: Impact of load on flow size distribution.

4.4.1 Impact of load

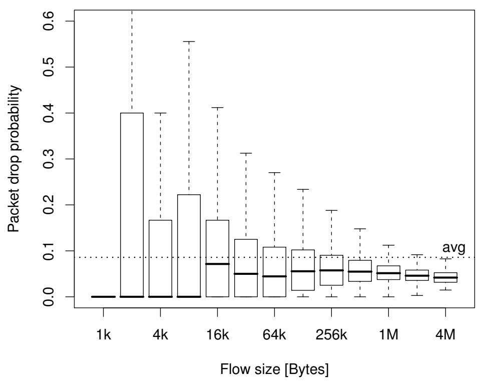
We start by studying how different flow sizes are impacted by losses for different offered loads. For each individual flow, the relevant information is not the overall loss rate but the fraction of its packets that have been dropped. Therefore, we compute packet loss rate for each flow as the fraction of packets that were dropped divided by the total number of packets sent by the sender.



(a) Low load



(b) High load



(c) Very high load

Figure 4.11: Per flow-size packet loss probability for different loads (128 packets buffer).

Figure 4.11 shows the per-flow packet drop probability (y-axis, using box-plots[5]) across different TCP flow sizes (x-axis) and for low, high, and very high offered load. Flow sizes are binned into logarithmic sizes. The buffer size is 128 packets and flow sizes are Pareto

[5]Box-plots show the minimum, the percentiles 25, 50, 75, and the maximum.

distributed with $\alpha = 1.2$. Different flow size distributions and larger buffers show similar behaviors and are omitted due to space limitations.

The total packet loss probability in this scenario is rather small with 1%. However, the loss is not distributed evenly across all flows. Under low load (Figure 4.11(a)), we observe that flows with sizes from $512K$ to $32M$ suffer from higher loss rates compared to other flow sizes. The careful reader might remember that in Section 4.3.2.2 we observed that UDP traffic observes higher loss rates than TCP when the load was low and buffer size large. This is coincidentally the same situation as in Figure 4.11(a). The UDP flow sizes across the experiments fall in the range of unlucky flows that may actually suffer from higher losses than the total packet loss probability.

When the link utilization is high (Figure 4.11(b)), a subset of the small flows suffer from larger packet loss probability than the set of larger flows. Note, most of the small flows still have a very small packet loss probability. Only some unlucky flows see more losses than the rest of the flows of a given flow size.

Under high load, the average packet loss probability across all flows sizes increases. Small flows tend to have a few unlucky flows that suffer from very high loss probabilities. For larger flows (larger than 16K) a larger fraction experience packet loss rates of roughly the same rate as the total packet loss rate. Even under very high offered load some happy flows do not observe significant losses.

We thus conclude that whatever the offered load is the observed packet loss probability of a single flow is unlikely to be representative of the total packet loss rate. Even very large flows, which one may expect to better sample the overall loss rate, can observe packet loss probabilities that differ significantly from the overall one. In general, most flows will not observe many packet losses. However, some specific flows might observe unusually high packet loss probabilities—just as some of our UDP flows from Section 4.3.2.2.

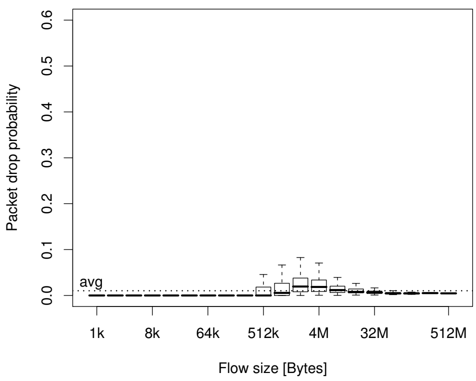
4.4.2 Impact of buffer size

Next, we examine the impact of buffer size on the packet loss probability for different flow sizes. Section 4.3 shows that reducing the router buffer size increases packet loss. We rely on the same flow size distribution as in the previous section (Section 4.4.1), but instead of varying the offered load, we now vary the buffer size for a low offered load.

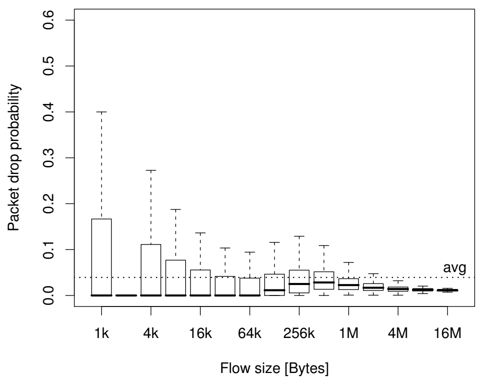
4.4.2.1 Flow happiness

Figure 4.12 shows the loss distribution across flow sizes for three different buffer sizes: 128, 64, and 32 packets.

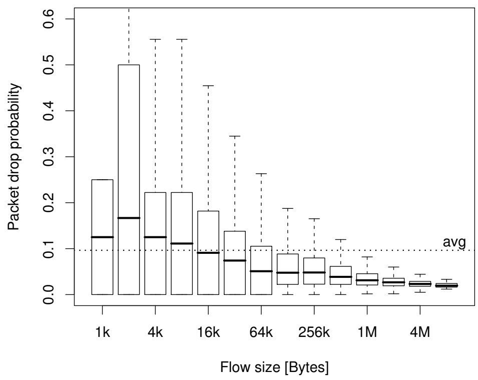
With a reasonably large buffer, only specific flow sizes experience unusually large packet loss probabilities, see Figure 4.12(a). When the router buffer size is small, smaller flow



(a) 128 packets buffer



(b) 64 packets buffer



(c) 32 packets buffer

Figure 4.12: Per flow-size packet loss probability for different buffer sizes (low load).

sizes are unhappy, and suffer from high packet loss probabilities. Contrary to scenarios with the high offered load, small buffers do affect all flow sizes consistently.

Even though a high offered load seems to have a similar effect as limited buffer size on packet losses of small flows, load levels and buffer sizes do affect flows rather differently. High offered load creates variability in the way losses are distributed across the different flow sizes—most flows do not see as high a loss rate as the overall loss rate indicates and there are only a small number of very unlucky flows (especially among the small ones) that suffer from unusually high losses. This is the case since high offered load with large buffers does allow some flows to send large bursts. However, these will be dropped when a full buffer is encountered.

In the case of very small buffers, all TCP flow sizes are affected by losses on average, because small buffers cannot absorb large packet bursts. Therefore, a very limited fraction of TCP flows have a chance to send packets without observing losses, no matter how low the load is.

A closer look at the plots in Figures 4.11 and 4.12 reveals another difference between high offered load and small buffer sizes. Under high load and large buffer sizes, larger flows tend to observe packet loss probabilities that are closer to the overall average. Under small buffers and low load, it is the small flows that tend to observe packet loss probabilities closer to the overall average. This suggests that under high load, only large flows that last long enough have a chance to properly sample the actual loss probability. When small buffers are the bottleneck, large flows do not representatively sample the actual losses inside the buffer because the buffer limits their TCP congestion window.

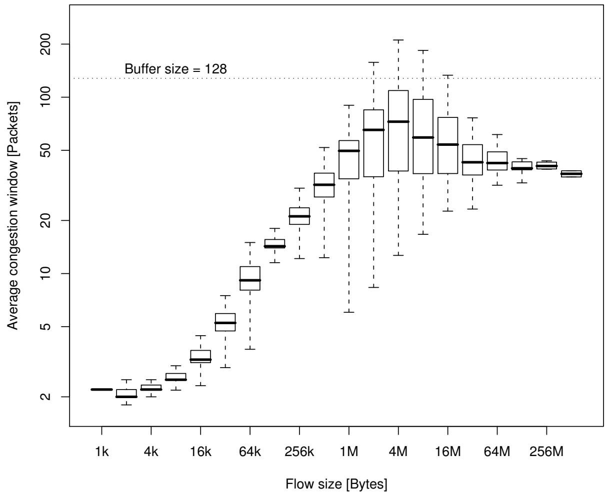
Similar to the high offered load case, reducing the buffer size increases the probability that some small flows will observe high packet loss probabilities. Furthermore, most flows observe much smaller packet loss probabilities than the global one and a limited fraction of the flows observe unusually high packet loss probabilities.

4.4.2.2 Buffer size and congestion window

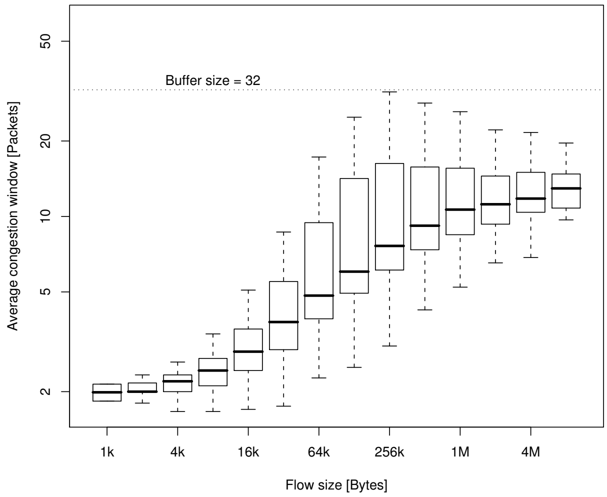
Under low utilization and with a large router buffer, we cannot expect that TCP reaches congestion avoidance for small flows. The average congestion window size for each TCP flow, see Figure 4.13, confirms this intuition. The x-axis of Figure 4.13 shows the TCP flow sizes while the y-axis shows the distribution of the average TCP window size over the flow lifetime using a box-plot. The top/bottom plot of Figure 4.13 corresponds to a buffer size of $128/32$ packets. Flow sizes are again Pareto distributed with $\alpha = 1.2$.

Only large flows manage to reach an average window size of the same order of magnitude as the buffer size. Except for very small flows and very large ones, the average window size grows with the flow size until it reaches values in the order of the buffer size. Note, the congestion window can take values as large as twice the buffer size before TCP will be signaled that congestion occurred at the buffer. Interestingly, those very flows for which the TCP congestion window grows beyond the buffer size are those who observe the unusually high packet loss.

When the buffer size is small, e.g., 32 packets, as in Figure 4.13(b), we observe that the average TCP congestion window is limited by the buffer size. Given that the throughout achieved by a TCP flow depends highly on the congestion window, small router buffers limit the performance of TCP even when there is bandwidth available along the path of the flow. In that case, the actual throughput that can be achieved by a TCP flow is much lower than what one might expected from the bandwidth delay product limit, which is 3025 packets in our case. This phenomenon has been observed in residential traffic [127]. We note, that similar observations hold for other flow size distribution including an exponential flow size distributions.



(a) 128 packets buffer



(b) 32 packets buffer

Figure 4.13: TCP congestion window against flow size (low load).

4.4.3 Time dynamics of packet loss process

Internet traffic has been shown to be bursty, exhibits scaling properties [74, 110] and is non-stationary [58, 113]. Contrary to what has been assumed in models of TCP [42] about the randomness and stationarity of losses, we would expect that the loss process actually exhibits non-trivial properties over time. Therefore, we examine in this section the loss process across time.

We start with the low offered load scenario and vary the buffer sizes. Figure 4.14 shows the scaling plots computed on a timeseries of the packet loss process at a time resolution of 1ms. The scaling plot [32] shows, at each time-scale j, the energy contained in the wavelet coefficients $(y(.))$. Since the timeseries has a time resolution of 1ms octave 1 corresponds to a time-scale of 2ms. Each successive octave j offers twice as coarse a resolution as the previous octave. The typical RTT, around 150ms, corresponds to octaves $7-8$. As the loss process might differ across time we use a 3D version of the scaling plot [190]. It shows the evolution across time of the scaling plot computed across over-lapping time intervals. Each time interval over which a single scaling plot takes a 60 seconds time series. We compute a scaling plot every 30 seconds in order to give a smoother look to the 3D plot.

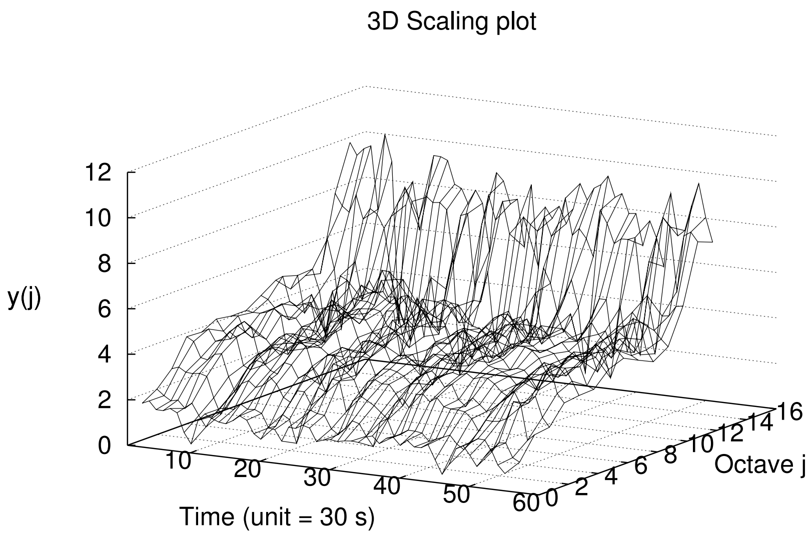
For a large buffer size (Figure 4.14(a)), e.g., 256 packets, the 3D scaling plot indicates that the loss process exhibits irregularity over time (varying level of consecutive scaling plots), and some possible scaling over time-scales below a typical RTT. When the buffer size is very small on the other hand (Figure 4.14(b)), e.g., 32 packets, we observe a flat scaling plot for time-scales below the typical RTT, indicating an uncorrelated process. For octaves larger than the typical RTT irregular behavior appears also in this packet loss process.

Increasing the load has a significant effect on the loss process, as can be seen on Figure 4.15 which again shows a 3D version of the scaling plot. When the offered load is high the loss process exhibits scaling properties at time-scales below the typical RTT.

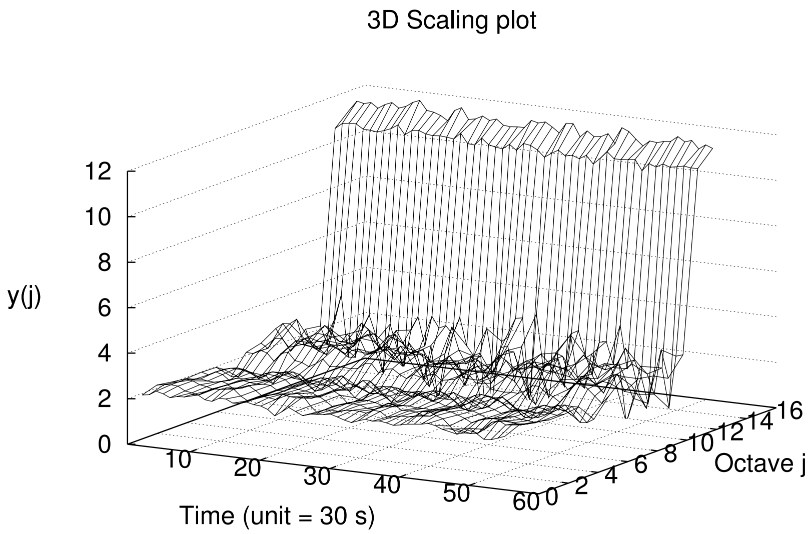
The difference in the scaling properties of the loss process under small buffers and high load further confirm that two different effects take place. Under high load and large buffers, TCP is allowed to be bursty by the large buffers but has highly variable losses, therefore the scaling. When router buffers are small on the other hand, TCP has not much chance to be bursty, and will therefore generate losses that are uncorrelated.

4.5 Discussion

Buffer sizing: In principle, our work should be comparable to previous studies that have investigated the implications of buffer sizing schemes [44, 47, 48, 72]. Unfortunately, buffer sizing studies make assumptions about the nature of the traffic properties, i.e. TCP senders need to pace their traffic. These assumptions make sense if the considered traffic is similar to what is observed in very high capacity links where a very large number of flows are multiplexed.



(a) 256 packets buffer



(b) 32 packets buffer

Figure 4.14: Impact of buffer size on the packet loss process (low load).

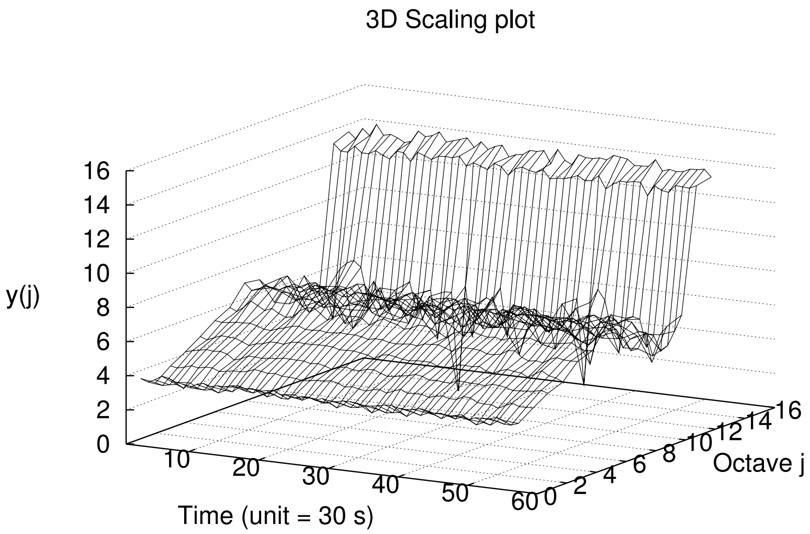


Figure 4.15: Impact of high load on the packet loss process (256 packets buffer).

In this chapter, we do not make any assumptions about traffic properties, and we have a limited number of concurrent flows. Therefore, the burstiness of our traffic is representative of access networks, where a limited number of users are aggregated.

On the one hand, we confirm that relying on router buffers much smaller than the bandwidth-delay product is possible without impacting the ability of TCP to utilize link capacity. On the other hand, given that the TCP congestion window is upper bounded by the buffer size, reducing buffer sizes must be done carefully not to impede on the throughput of TCP.

Applications performance: Our study highlights the importance of understanding the flow-level properties of the traffic, e.g., packet loss, under different network conditions, and their consequences on applications performance. For example, an application like Web that usually transfers limited size objects, may suffer from very high losses due to high load, leading to unacceptable Quality of Experience. Bulk data transfer applications that exchange large files might rather be impacted by small router buffers that limit the TCP throughput.

4.6 Related work

In the past many researchers have studied the correlation between packet losses on Internet paths, e.g., [52, 144, 203]. These approaches typically rely on active measurements for sampling the path properties of the data plane, e.g., send probes every tens of milliseconds. Due to this sampling, the actual loss process has to be inferred from observed losses experienced by the probes. While such an inference process might accurately estimate the average path loss observed by a flow on a given path, understanding how the losses are distributed among the flows that share a router buffer requires to observe the traffic at the buffer. This is the approach that we take in this chapter.

Note, previous work typically assumes that each probe samples the loss process independently and is unbiased by: (a) the packet stream which is used for the probing which may be UDP or TCP, as well as (b) the size of the flows used for the measurements. We, in this chapter, show that both assumptions are questionable: the choice of the flow size as well as the transport protocols impacts the observability of the loss process.

The literature is not necessarily focused on the loss process itself, but may also consider some of its implications. For example, Sommers et al. [185] focus on measuring loss, delay, and jitter from active measurements to check compliance with Service Level Agreements (SLA). While such measurements are likely appropriate for referring to the overall quality of the data plane as seen by particular flows, they do not refer to the actual loss process inside the router buffers as we do in this chapter.

Another area of related work regards the sizing of router buffers, e.g., [44, 47, 48, 72]. While sizing router buffers is not the focus of this chapter, our work sheds light on the consequences of different buffer sizes on the performance of individual flows and thus to application performance. Buffer sizing relies on assumptions about the number of flows that share a link as well as on the traffic burstiness. We do not question those assumptions, but show in this chapter that the buffer size limits the size of the TCP congestion window. This in turn affects the throughput achievable by individual flows.

4.7 Summary

Through controlled experiments, in this chapter we study the relationship between several parameters, including load, router buffer size, and flow size distribution, on the properties of traffic and the loss process.

Our sensitivity analysis reveals that small buffers have a deep impact on the ability of TCP to use the link capacity. We confirm that both high load and small buffers lead to high packet losses. However, small buffers have a much higher impact on losses than high load.

Surprisingly, we find that packet losses do not affect all flows similarly. Irrespective of the network load and the buffer size, there are few unhappy flows, especially small ones, that observe unusually large losses. On the other hand, most flows, especially large ones, are happy and do not observe high losses compared to the global loss rate. Furthermore, very few flows actually observe a loss rate similar to the average loss rate. Therefore, any single flow is very unlikely to observe the global packet loss process.

Finally, our study of the packet loss process reveal that it can exhibit scaling properties under high load as well as significant irregularities under large buffer sizes. When the buffer size is very small, the loss process is uncorrelated at time-scales below the typical RTT.

5

Flow performance in the wild

5.1 Overview

With controlled experiments in a testbed setup only a limited number of network conditions can be reproduced and analyzed. In the Internet, a plethora of network conditions exists and reproducing all such conditions in a testbed setup is a formidable task due to a number of reasons. The difficulty stems from the scale of the Internet, diversity in the network equipment, dynamic interactions between the traffic of different applications protocol, disparity in network topological structures, differing users behavior, and heterogeneity in the access networks.

Recall from Section 1.1 that application protocols mix has evolved towards more HTTP based applications [117, 127]. However, P2P applications still carry, although less, significant traffic volume. These applications deliver content to users, by relying, e.g., on the ADSL upstream capacity of other users, which is often limited. The popularity of social networking has given rise to User-generated content (UGC) which also creates congestion in the upstream while pushing content to the Internet. Despite the efforts of network operators to cope with the capacity demands of the users, congestion still exists and it will continue to exist in the Internet and impact user experience.

Understanding where and how bottlenecks impact flow performance is crucial for application providers, Internet service providers, and end-users. Recent studies from Google, Amazon, Yahoo, and Microsoft have demonstrated that few milliseconds difference in web-performance impact business value [186]. To account for all aspects of the Internet, in this chapter, we now focus on exploring flow performance from the real Internet traces. We seek to investigate the impact on flows as a result of network congestion, applications behavior, limited DSL capacities, and TCP loss recovery mechanisms.

We show that flows of different sizes from various applications receive different service quality. We devise a methodology that targets *individual* flow performance of different applications using passive traces collected at multiple vantage points. Our work differs from prior research in that we consider the following performance metrics: TCP retransmissions, out-of-sequence packets, throughput, and RTTs based on different classes of flows according to their size, hereafter referred to as *flow-classes*. We use TCP retransmissions as a key performance metric as any retransmission in a flow indicates either packet loss or high latency triggering a timeout event. Our aim is to highlight that flows can experience severe degradations that would be considered unacceptable by today's end-users. Our key takeaways and lessons learned in this chapter are:

Lesson 1 *Packet retransmissions vary significantly across different flow sizes. For example, retransmissions for flows smaller than 128KB range from less than 2% to as much as 40%. Unexpectedly, some large flows do not experience any retransmission or reordering.*

Lesson 2 *P2P (e.g., Bittorrent/eDonkey) and unclassified traffic dominates in terms of packet retransmissions. The average retransmission rate per flow for this traffic is much higher than for other applications, especially in the upstream direction. This confirms the anecdotal evidence of P2P applications creating bottlenecks by monopolizing the upstream capacity. At the same time, very large P2P flows perform reasonably well without incurring high retransmissions.*

Lesson 3 *Irrespective of the considered applications, most retransmissions in the upstream direction take place within the network segment between the DSL customer and monitoring point, whereas retransmissions in the downstream direction are due to the rest of the Internet.*

Lesson 4 *HTTP and P2P applications differ in how they leverage loss recovery mechanisms: large HTTP flows are able to more easily recover from losses than P2P. For small flows on the other hand, both HTTP and P2P are similarly unable to take advantage of TCP loss recovery mechanisms. We observe significant timeouts for small flows, especially P2P ones.*

The remainder of the chapter is structured as follows: In Section 5.2 we describe our methodology. Section 5.3 presents details about the data sets we use. We present our results about retransmissions across flow sizes in Section 5.4. In Section 5.5, we study the behavior of different applications. Section 5.6 studies which part of the network is responsible for retransmissions. Section 5.7 investigates the use of the different TCP recovery mechanisms. We discuss related work in Section 5.8 and summarize our work in Section 5.9.

5.2 Methodology

To understand the traffic characteristics, it is common to summarize the data at the flow level, e.g., using classical 5-tuple flows based on source and destination IP addresses, port numbers, and protocol. This is not sufficient for us since we need detailed information, in particular about TCP retransmissions, TCP loss recovery mechanisms, RTTs, throughput, and DSL access speed. In addition, we want to use an application level detection mechanism that is not purely port based. In the following, we discuss our approach for gathering all necessary information.

5.2.1 Annotated flow summaries

We use the network intrusion detection system Bro [155] as it provides comprehensive analysis capabilities of TCP connections and is able to handle large datasets. In this context, a *TCP connection* refers to a bi-directional TCP communication which starts with the arrival of TCP SYN packet and terminates with FIN/RST packets by either side. Statistics reported by Bro include start time, durations, originator IP and destination IP, originator port and destination port, application protocol, direction, TCP state, additional flags (e.g., to indicate payload data in both directions), payload bytes and packet counts in both directions, as well as a round-trip-time (RTT) sample. The state information captures the state of the hand-shakes. Both the initial three-way hand-shake as well as the final connection closing hand-shake. The RTT sample is an estimation of the round-trip-time as obtained from the initial TCP hand-shake using a similar methodology as reported in [109]. Payload bytes are accounted as seen in the packets on the wire rather than being estimated from the TCP sequence space numbers.

To determine the application protocol of the connection we rely on Bro's Dynamic Protocol Detection (DPD) [67] mechanism. Bro includes a wide range of protocol analyzers including HTTP, Bittorrent, edonkey, FTP, POP3, SMTP, etc. These analyzers detect application protocols by parsing the connections byte stream and matching it to multiple application signatures in general. For each matching signature a specific analyzer is started which verifies that the bidirectional communication is consistent with the application layer protocol. If this is the case the connection is annotated with the appropriate protocol label.

While Bro is capable of detecting a large base of application protocols, we choose seven main categories of applications for our analysis. Our first four categories are HTTP, Bittorrent, edonkey, and SSL. In addition, we group together traffic which is identified by DPD but does not belong to the first four categories together into the group OtherDPD. This includes traffic from protocols such as FTP, POP3, SMTP, and IRC. The well-known category includes traffic on well-known ports. All the remaining traffic is in Un-classified. A summary of the available information can be found in Maier et al. [127].

We convert each connection into two half connections, one in each direction since we want to do separate analysis for both directions. In the following, we refer to these half connections as *flows*. These flows are then annotated with richer meta-information than typical for standard 5-tuples.

One of the standard problems with analyzing packet level traces is the question about how to deal with connections that start before or end after the end of the trace. We focus on those flows for which we see a proper three way hand-shake as well as a connection tear-down either via FIN or RST. This has the following additional advantages. We will only consider flows from TCP connections for which we can observe both directions. Moreover, scans as well as SYN, SYN ACK attacks are excluded. It also reduces edge effects.

We typically consider two different observation periods: full days and hours. For the full days the impact of the edge effects is relatively small. To reduce them for the hour long periods we include all flows that started within the considered time bin. However, for flows that last longer than our time period, we filter out the part of the flow outside the considered bin, i.e., we do not include packets that are sent outside of the time period.

5.2.2 Flow classes

It is well-known [64] that Internet flow sizes are consistent with heavy-tailed distributions. Thus, we use logarithmic classes, referred to as *flow-classes*, for binning flows based on their payload bytes. We bin flows into flow-class i, so that for all flows within the flow-class i we have $2^i < payloadbytes \leq 2^{i+1}$ for $i = 0, 1, 2..n$. The largest flow-class also contains flows larger than 2^n. By analyzing flows separately for each size-based class, we gain insight about the relative behavior of flows across flow-classes. We typically start with a flow-class of 1KB and go up to a flow-class of 1GB. In addition, we, for some analysis, use the application type to first separate, e.g., HTTP flows, and then look at the performance of flows within each of the logarithmically sized flow-classes.

Part of the motivation for looking at flows across different size-based classes is that different flow-classes may be dominated by different types of flows. For example, some types of video objects have median sizes of 265KB, 802KB, and 1743KB [89]. Likewise, most of the Google products have flows in the range of 4-16KB [69]. Moreover, recent studies about the changing nature of website complexity [56] have shown that overall median webpage sizes for short, medium, and long web pages has grown to 40KB, 122KB, and 286KB respectively. Moreover, when a user observes good application performance, he may be tempted to access larger objects and thus generate bigger flows. If the performance is impaired on the other hand, e.g., due to retransmissions, he may restrict himself to smaller objects, e.g., a lesser quality image, which corresponds to smaller flows.

5.2.3 Retransmissions

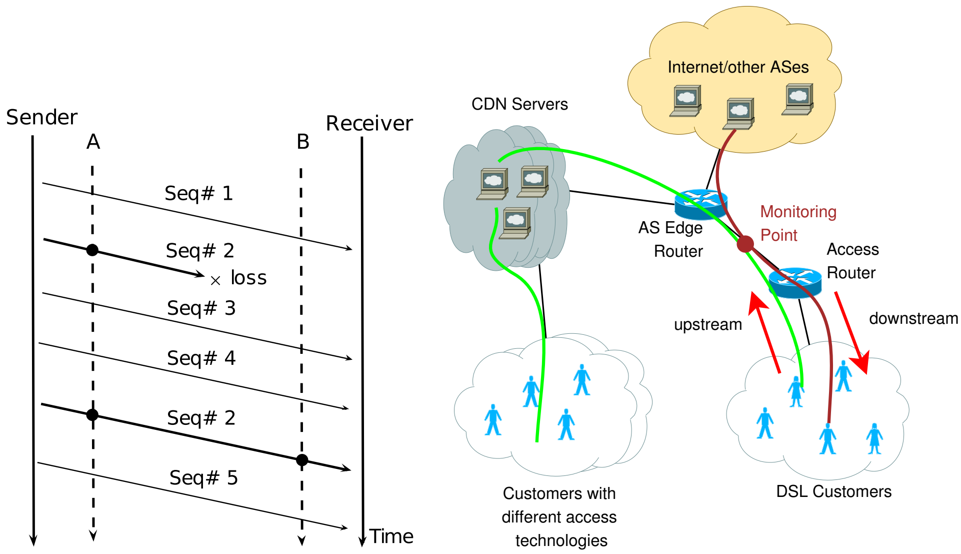
While TCP is designed to be robust against packet losses and/or reordered packets, flow performance suffers significantly in their presence, see, e.g., the report by Padhye et al. [150]. Indeed, a significant amount of research has focused on making TCP even more robust, e.g., [41, 49–51, 118]. Therefore, ideally, we want to include both the number of packet losses and the number of reordered packets in our performance metrics. However, it can be very hard to distinguishing between them [107]. A first approximation is to consider out-of-sequence packets. The next step is to check if they are due to retransmissions and reordering. Finally, we are using the methodology by *tcpcsm* [39] to identify the cause of the retransmission.

Part of the complexity of identifying TCP retransmissions from a passive packet level trace is that the trace is collected in the middle of the network and not at any of the endpoints where the TCP state is available as well. Thus, the challenge is to infer the TCP connection state, including continuous estimation of RTT, and/or congestion window. Moreover, due to multi-path routing and/or load balancing techniques not all packets may be seen at the monitor.

We again rely on Bro to implement our mechanism for detecting of out-of-sequence and retransmitted packets based on ideas by Paxson [154]. The advantage of Bro is that it already tracks per connection state and scales to large data sets [68]. More precisely we classify out-of-sequence packets as follows:

- If a packet with the same sequence number as a previously seen packet is observed it is considered retransmitted. Note, we use the end sequence number of the data within the packet for sequence number matching, rather than the start sequence number. This ensures that all cases are excluded where the retransmitted packet transmits more payload than the original packet. Furthermore, we explicitly exclude window probe packets using this mechanism. We observe less than 1% retransmissions that are due to window probing and exclude them as these are related to bottlenecks with the application and not the network. However, we count SYN and FIN retransmissions as they indicate packet losses.
- If a packet with an unexpected sequence number according to the ACKs is detected, this indicates a hole in the sequence number space. In this case we compare the timestamp of the out-of-sequence packet to that of the highest sequence number seen so far for this connection. If this time difference is larger than the current minimum RTT estimate of the connection, then the packet is considered retransmitted. Otherwise, it is considered reordered. The underlying assumption is the expectation that reordered packets are not delayed longer than a round trip time. If the time difference is larger than the current minimum RTT estimate and it has a lower IPID, the packet is considered reordered.

This strategy allows us to detect out-of-sequence packets and separate them into retransmitted packets and reordered packets. In addition, it allows us to detect if the retransmissions



(a) RTX detection (b) Setup

Figure 5.1: Measurement setup

occur before the monitoring point or after the monitoring point for both directions. Figure 5.1(a) explains the detection mechanism with the help of two observation points A and B. In the example, packet with seq#2 is lost and retransmitted after a timeout event. Note, we do not show ACKs in the reverse direction due to simplicity. If a packet is dropped after the monitoring point or times out, e.g., due to high delays, then monitor A will see the packet twice, implying that the path segment between monitor and the receiver is responsible for the loss event. On the other hand, if a packet with a lower sequence number is seen at monitor B after a min-RTT spacing, then this implies the path segment between the sender and the monitor is responsible for the packet loss. Using this methodology, we distinguish if the retransmission is due to the sender local or the receiver segment.

5.2.4 Retransmission cause

We chose to not augment Bro, as it would require to implement an analyzer which knows about the various different variants of TCP retransmission mechanisms of the various different TCP stacks of the operating systems. Rather, we rely on the tool *tcpcsm* [39]. Tcpcsm's basic design strategy for detecting retransmission time-outs (RTO) is based on the elimination principle. Tcpcsm tries to detect all other possible methods of retransmissions such as fast retransmissions, fast recovery, SACK recovery, F-RTO recovery, spurious retransmissions, etc. The remaining retransmitted packets are declared as RTOs. This approach has the advantage of not requiring to keep track of the separate TCP state machines of different operating systems. Fast retransmissions are detected after three explicit duplicate ACKs. Subsequent retransmitted packets are detected as fast recovery. If a retransmission occurs as a result of a SACK gap, such retransmission is recorded as a SACK retransmission. The detection of reordered packets depends on observing lower sequence numbers and lower IPIDs. Unnecessary or duplicate retransmissions are also recorded separately, based on the observations of ACK packets. Tcpcsm relies on 13 different tests to classify the state of retransmitted packets. Thus, tcpcsm improves upon previous work [169] of capturing TCP state machine from network traces and is an excellent tool for our purpose.

Via extensive manual inspection, we checked tcpcsm's loss recovery detection mechanisms. For each packet that we identified as retransmitted using the above algorithm within Bro, we are able to find the possible retransmission mechanism via tcpcsm. Each packet identified only by tcpcsm as being retransmitted either belongs to a connection which we did not consider, e.g., if we did not observe both the initial as well as the final hand-shake, or it is among those that we classify as out-of-sequence. A small fraction, well less than 0.5-2.2% of the packets, identified as retransmitted by tcpcsm are not considered retransmitted by Bro. Manual inspection has shown that among the reasons for the difference is that our methodology within Bro cannot detect packet losses at the end of a flight as these are not out-of-sequence. Moreover, Bro relies on an estimate of the minimum RTT for its separation of retransmitted and out-of-order packets.

5.2.5 Flow performance metrics

We use throughput, RTT, out-of-sequence, and retransmission packet and byte rate as key flow performance metrics. In the following we present how we compute them. (1) We calculate throughput as the ratio of the payload bytes and the duration of the flow. (2) As round-trip-time estimation we use the one from the initial three way hand-shake as calculated by Bro. (3) The out-of-sequence packet rate per flow is calculated as $\frac{\sum(retransmit+reorder)}{totalpackets}$. (4) The retransmission rate per flow is calculated as $RTX = \frac{retransmit}{totalpackets}$. (5) The byte retransmission rate per flow is calculated as $BRTX = \frac{retransmit}{totalbytes}$. Since the results for BRTX are similar to those for RTX we only report the latter.

5.3 Datasets and terminology

In this section, we describe the three datasets used in the chapter. See Table 5.1 for an overview of our traces. In this chapter, we mainly focus on the MAR-10 DSL trace. However, we have verified our results across all other traces and will point out differences where applicable. In general, we report our results for the full 24 hours traces as well as for selected hour-long time periods from different times of the day. This allows us to check for time of day effects.

Table 5.1: Summary of properties of anonymized traces.

Name	Type	Time	Size	Duration
AUG08	packet (full)	Aug 08	> 4TB	24h
APR09	packet (full)	Apr 09	> 4TB	24h
MAR10	packet (full)	Mar 10	> 4TB	24h
MAWI	packet (headers)	Apr 10	> 135GB	43h
CDN	conn. logs (sampled)	Mar 10	> 50GB	2 weeks

5.3.1 Residential broadband ISP traces

Our first data set consists of anonymized packet level traces of residential DSL connections collected at an aggregation router inside a large European ISP. This broadband aggregation router situated in the backbone network of the ISP is a gateway for more than 20,000 DSL customers to the Internet. Data is collected through the help of monitors equipped with Endace cards and is immediately anonymized. The monitor observes traffic from customers with varying line speeds that range from 1200/200 Kbps (downstream/upstream) to 17000/1200 Kbps. In our data set, the distribution of access speed for lines with downstream speeds 1200, 2300, 3500, 6500, and 17000 Kbps is 15, 20, 22, 30, and 8% respectively. From this monitor, we have collected packet level traces in 2008, 2009, and 2010 each covering a 24 hours period. Overall, the traffic pattern follows a diurnal pattern and never exceeds a link utilization of 45% during the peak hour.

We call traffic that is sent by DSL customers *upstream* and traffic which is received by the DSL customers *downstream*, see Figure 5.1(b). Similarly, we refer to the network segment between the DSL customer and the monitoring point as the *local side*, whereas the *remote side* refers to the network segment between the monitor and the rest of the Internet.

Using Bro with DPD we classify the traffic according to application protocols into seven categories. In our traces we find, consistent with the results of Maier et al. [127], that HTTP is the dominant protocol with a traffic share of more than 60% in the downstream direction. In the upstream direction, HTTP has a share of 30%. However, this varies significantly across time and can go as low as 10% during off-peak hours in the early morning. The total traffic from P2P—Bittorrent, e-Donkey—and un-classified is less than 25% of the overall traffic. However, it in the upstream direction traffic is dominated by P2P (Bittorrent 20%) and un-classified traffic (22%). The relative share of P2P and un-classified traffic increases by additional 10-20% during off-peak hours for both downstream as well as upstream.

5.3.2 Content distribution network logs

Our second data set consists of connection level logs from the servers of one of the largest content distribution networks (CDNs). We specifically select servers which are serving customers of the same large European ISP from which we gathered the DSL traces. The connections include both those by DSL users as well as all other customers of this ISP. The data within the logs is obtained via kernel level monitoring on the CDN caches and include low level statistics such as total packets, bytes, retransmitted packets and bytes, RTTs, and durations for each TCP connection. Due to the huge data volume these logs are only generated for sampled connections. Statistics for all flows are maintained in the kernel, and once the sampling mechanism is triggered, statistics for that flow are recorded to disk. Note, contrary to the previous data, this data is single-sided. It captures only the HTTP traffic in the direction from the CDN to the customer. To ensure that we are only considering flows to customers, we filtered out internal connections between CDN clusters.

5.3.3 MAWI trans-pacific traces

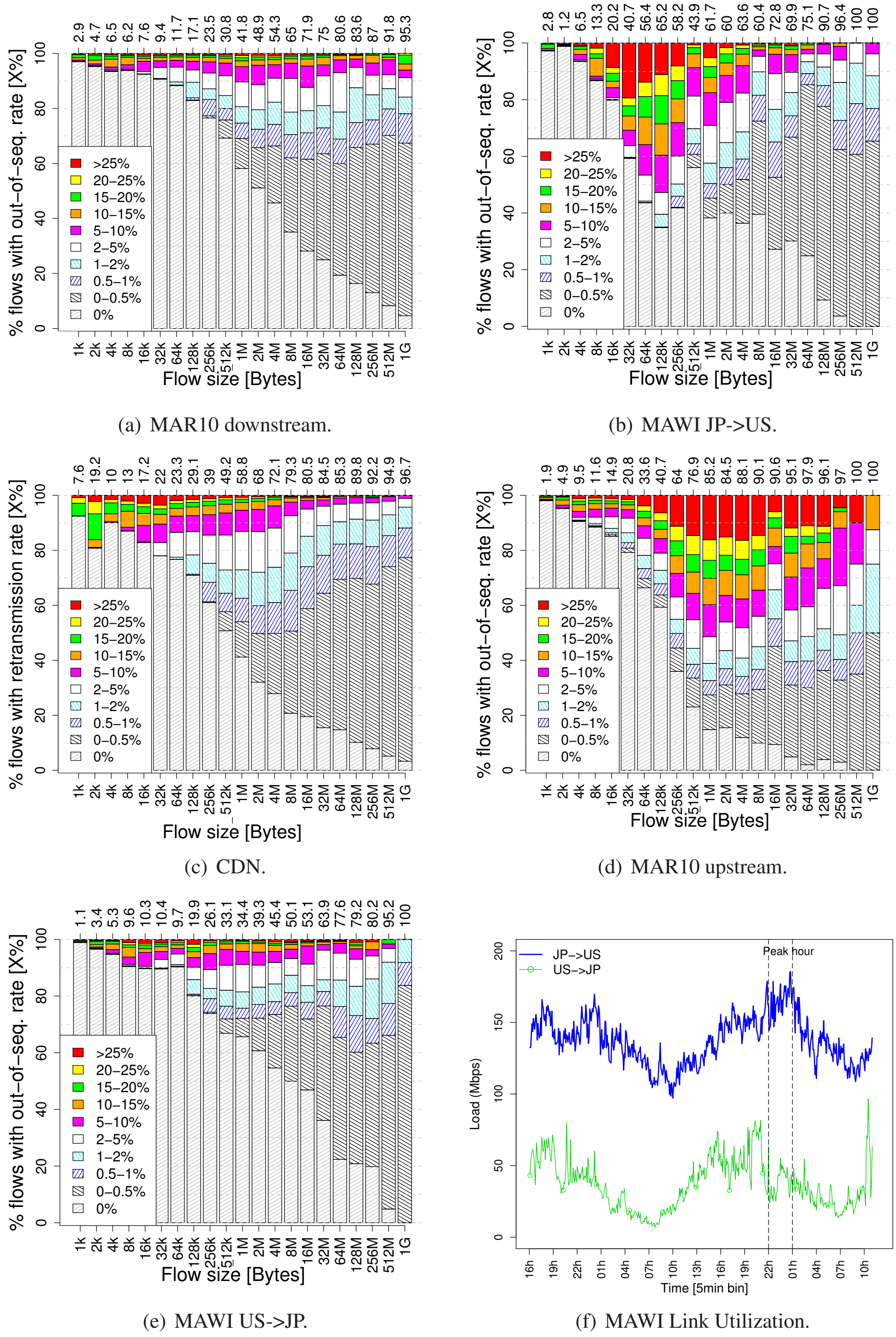
The third data set consists of a 43 hour anonymized packet header trace from the 150 Mbps trans-pacific transit link between Japan and the US from the WIDE backbone [131]. While this link is in principle rate limited it has some additional headroom (with support of the operating ISP). The utilization of the monitored link differs from the other traces in that it is operated at high link utilizations. During peak hours we see a utilization of almost 100% (sometimes even more during relatively short time periods) and 75% at off-peak hours. This high load was one of our motivations to consider this dataset. Due to lack of payload information, we cannot use DPD based application detection. Therefore, we rely on port based application identification. HTTP is again the most dominant protocol with more than 50% of the traffic. 10% of the traffic cannot be classified using well-known ports. This environment has a relative large fraction of UDP traffic with 20-25%.

5.4 Flow size matters

We begin our study by asking the question if out-of-sequence packets are distributed evenly across flow size classes. In this section we focus on out-of-sequence packets as they are fast to compute and, as we will show later in the chapter, are a good approximation for retransmissions.

5.4.1 Out-of-sequence packets—Overall

Overall, we observe less than 1.5% out-of-sequence packets across all traces. This is comparable to previous results [69, 158] which observe retransmission rates in the order of 1-2.8%. More precisely, we see 1.2%, 1.2%, and, 1.5% out-of-sequence packets for MAR10, MAWI, and CDN.



(a) MAR10 downstream.

(b) MAWI JP->US.

(c) CDN.

(d) MAR10 upstream.

(e) MAWI US->JP.

(f) MAWI Link Utilization.

Figure 5.2: Out-of-sequence packet rate per-flow for full trace / Link utilization.

Out-of-sequence packets are only seen by 9.2%, 9.9%, 16.91% of the connections in from MAR10, MAWI, and CDN. A large number of connections therefore do not see any out-of-sequence packets. This implies that some connections see more out-of-sequence packets than the average. For example, 1.4%, 3.1%, 3.5% of the connections have more than 20% out-of-sequence packets for MAR10, MAWI, and CDN. Out-of-sequence packets are therefore not equally distributed across all connections, and already a first answer to our above question is negative.

At first glance, this may be counter intuitive. How can a TCP connection make any progress with that many out-of-sequence or retransmitted packets? We manually checked more than 50 connections and verified that those packets identified as out-of-sequence or retransmitted packets were indeed valid and that the connection still made progress. Among the reasons for the large number of retransmissions, is that, if the congestion window is large, loosing a single packet can lead to a large number of retransmissions. Moreover, repeated timeouts also lead to a significant number of retransmissions. Thus, iterations of timeouts with successful transmissions can easily lead to large retransmission rates while still successfully transferring data. This is counter intuitive given that observations [60] have shown that with packet loss rates larger than 5%, e.g., as observed in some wireless networks, TCP throughput decreases in such a way that TCP is unable to use the network capacity.

5.4.2 Flow size—Motivation

In the past, bulk transfers and their performance have attracted a lot of attention from operators and researchers, for multiple reasons such as optimizing network bandwidth usage [179] to new protocol design [82, 189]. However, while bulk flows contain the majority of the bytes, most of the flows are short [64]. We note, that in comparison to the attention that bulk flows have received, short flows have received almost none. Yet, it is known that the performance such flows receive can be crucial for the experience of the user. Therefore, some researchers have proposed to give short flows priority over long flows [45]. Still, there is a general belief that short flows do not face as much trouble as long flows since they are not subject to congestion control.

In this section, we verify this belief that short flows do not face much trouble based on our traces. Figure 5.2 shows the fraction of out-of-sequence packets for each environment. Each plot uses logarithmic flow binning according to flow size (see Section 5.2.2). For each bin, we compute for all flows within the bin the percentage of out-of-sequence packets. We then use another binning to show what percentage of flows within a given size bin, have a percentage of out-of-sequence packets that falls within the bin range. This data is then plotted as a stacked barplot with a separate bar per flow size class. Within this bar we show the fraction of flows with an out-of-sequence range larger than 25% at the top and the fraction of flows with no out-of-sequence packets at the bottom. Thus, the y-axis shows, for each flow size bin, the cumulative percentage of flows with an retransmission rate of at least y. In addition, the numbers on top of the bins indicate the overall percentage of flows within the bin that have out-of-sequence packets.

The plots in Figure 5.2 are visual confirmation that the answer to our question is that **out-of-sequence packets are not evenly distributed across flow-classes.**

5.4.3 Flow size—DSL access

ADSL provides broadband Internet access and typically has highly asymmetric bandwidth at the access. These networks generally rely on an over-subscribed access network, an over-provisioned backbone, and a usually under-utilized home network. As such one may expect that small flows manage to sneak through while large flows may suffer from occasional performance problems.

We start with Figure 5.2(a), which provides the results for the DSL downstream direction. As expected, the fraction of flows with no out-of-sequence packets decreases as the flow size increases. However, Figure 5.2(a) highlights that between 2.9 and 17.1% of the flows smaller than 128KB experience at least one out-of-sequence packet during their lifetime. Indeed, the out-of-sequence rate for such flows is high, i.e., in the range of 2 to 15%, as compared to the roughly 4% observed by Jaiswal et al. [107]. We observe similar results in the AUG08 and ARP09 traces (not shown). For example, for the flow-classes 1-128KB, the percentage of flows with out-of-sequence packets varies between 5.6 and 23.3%, and between 3.3 and 20% respectively.

Small flows are not the only ones experiencing very low out-of-sequence rates—some large flows in the tail, i.e., the 1G flow-class, experience no out-of-sequence packets, even though TCP is designed to fully utilize the available network capacity by increasing its network usage until it experiences packet loss. Manual verification has shown that at least some of these are constrained by receive window limitations. About 60% of the flows larger than 1M see an out-of-sequence rate of less than 1%.

Most ISPs DSL offerings provide downstream to upstream ratios of 10:1, which roughly corresponds to the typical data ratios observed when browsing the Web. However, in times of user-generated content, the limited upstream speed can be a major hindrance. Thus, we next check the impact of out-of-sequence packets on the flows in the upstream direction, see Figure 5.2(d). The overall flow performance, as seen through out-of-sequence packets, looks vastly different than in the downstream case. First, the percentage of flows with out-of-sequence packets in the 1-128KB flow-classes increases to values between 1.9 up to 40.7%. Our 2008 and 2009 traces show similar values: between 3.3 and 56.1% and between 2.4 and 47.7%, respectively. The values for medium sized flows in the upstream direction are in between those for short and large flow-classes. More than 40% of the flows of size 256KB-8MB have an out-of-sequence rate above 5%, with around 15% of them experiencing more than 25% out-of-sequence packets. Indeed, the medium sized flows are among those experiencing the largest fraction of very large out-of-sequence packets.

We conclude with a more nuanced answer to our question: **Out-of-sequence packets are not evenly distributed across flow-classes - while small flows can have substantial num-**

ber of out-of-sequence packets, large flows may have none. Moreover, there are significant differences with respect to traffic direction.

5.4.4 Flow size—Congested backbone link

Next, we turn our attention on the MAWI traces to check if our results are not an artifact of the DSL environment. The MAWI traces allow us to study the flow behavior from the middle of the network—a trans-continental link. From Figure 5.2(f) which shows the bandwidth utilization over a 43h period, we see that the link in the JP to US direction has a high link utilization, above 82.5% for more than half of the trace period. WIDE backbone operator has confirmed that this is due to the presence of popular servers inside the WIDE backbone at that time, whereas the US to JP link has a utilization of 22.3% on average.

This environment also enables us to observe flow behavior in a setting in which we know that there is at least one heavily utilized link on the path. The corresponding plot for the flows in the JP-US direction is shown in Figure 5.2(b). We find that between 2.8 to 65.2% of flows smaller than 128KB experience out-of-sequence packets. These out-of-sequence values are comparable to those observed in the upstream direction in the DSL environment—an environment which can easily be subjected to congestion. Unsurprisingly, large link utilization results in larger and more severe out-of-sequence flow rates. Cross-checking the peak and off-peak hours highlights that part of the out-of-sequence packets correlate with the link congestion.

However, not all flow-classes are impacted by network load in the same manner. It appears that in particular the medium sized flows suffer. For the medium sized flows, we find that 30% of the flows in the classes 32KB-128KB have an out-of-sequence rate of more than 10%. In contrast, 60% of the large flows 64MB-1G have out-of-sequence rates smaller than 0.5-1%. This confirms the observations made in the DSL environment—even large flows may enjoy fewer out-of-sequence packets than some medium and/or small flows.

The US-JP direction (Figure 5.2(e)) shows qualitatively similar behavior to the DSL downstream direction (Figure 5.2(a)) with between 1.1 and 19.9% of flows smaller than 128KB experiencing out-of-sequence packets. We again see some very large flows without any out-of-sequence packet in both directions. Manual cross-checks of a subset of these flows again shows that they are congestion-window limited. This link is operated at a lower utilization and thus the lower out-of-sequence rates are not surprising.

We conclude with another more nuanced answer to our question: **Out-of-sequence packets are not evenly distributed across flow classes - while small flows may have a substantial number of out-of-sequence packets, large flows may have none. Moreover, there are significant differences with respect to link congestion level.**

5.4.5 Flow size—CDN's viewpoint

Next, we focus on a monitoring point close to the servers. Surprisingly, we again find that 7.6-29.1% of flows smaller than 128KB experience retransmissions, see Figure 5.2(c). (For this trace we use retransmissions.) Their number is slightly higher than for DSL (downstream) and lower than that of the congested link (JP-US). Among the reasons are that this CDN serves customers connected to the Internet through a wide variety of access links, including high-speed private networks, cable, mobile, as well as DSL. Overall, we note that the general shape and structure of the plot is similar to the others and thus flow behavior as seen from the CDN server perspective is similar to the one in other environments.

5.4.6 Summary

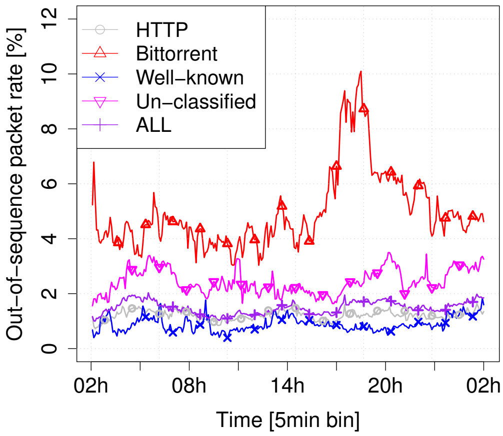
From examining traces from three different network locations we learn that:

- Flows of different sizes experience widely different out-of-sequence rates. While short flows often sneak through without suffering from out-of-sequence packets, a significant fraction of them suffer from out-of-sequence packets. This is likely to negatively impact the performance of the corresponding applications as the only loss recovery mechanism available for flows of this size is the time-out mechanism, based on default timeouts of three to six seconds.
- Congestion has an impact on retransmission rates, in particular on the medium size flows. Moreover, the comparison of the two traffic directions reveals that out-of-sequence packet rates in the DSL upstream direction are very similar those of a congested link in the middle of the network. A significant fraction of the mid-sized flows experience severe—larger than 25%—out-of-sequence rates. The results for the downstream direction of the DSL environment are comparable to those of the uncongested US–JP link. This indicates that the out-of-sequence packets are due to network bottlenecks.

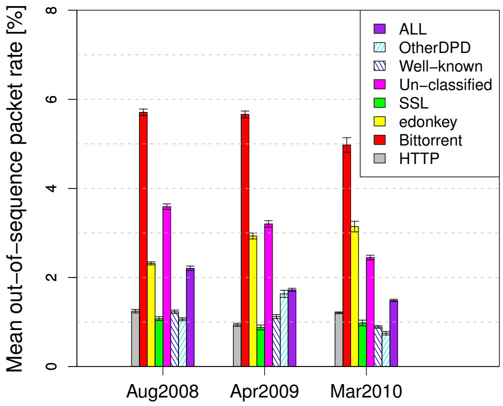
Assuming for the time being (see Section 5.5) that out-of-sequence packets are a good approximation for packet retransmissions, these insights are succinctly summarized as Lesson 1 in Section 5.1.

5.5 Applications and flow-classes

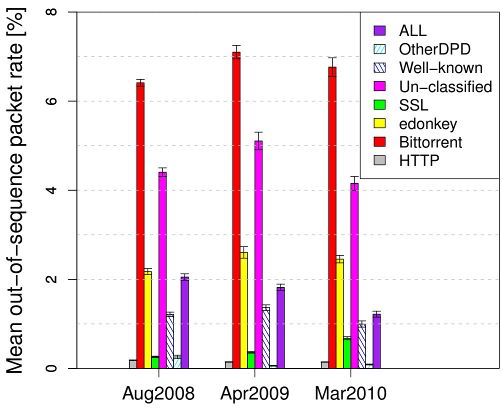
Next, we ask the question if out-of-sequence rates as well as retransmission rates change over time and if they are evenly distributed across applications. In particular, we want to understand if there are any significant differences between HTTP and P2P either across time or application protocol.



(a) Time series—MAR10 downstream.



(b) Histogram—MAR10 downstream.



(c) Histogram—MAR10 upstream.

Figure 5.3: Out-of-sequence packet rate per application across time and traces.

5.5.1 Across time—Application type

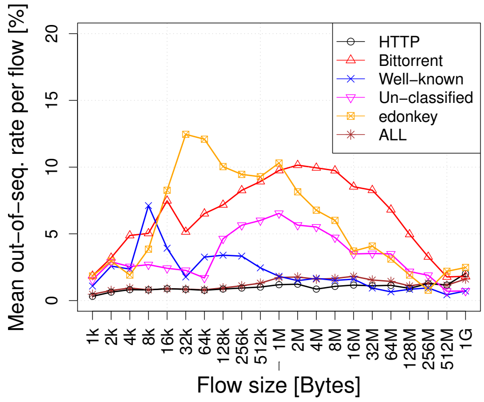
We start with Figure 5.3(a) which shows the average percentage of out-of-sequence packets for each 5 minute time bin across time, for the DSL downstream direction of the MAR10 dataset. We find that overall the mean percentage of out-of-sequence packets is relatively stable and small, between 1.29 and 1.66%. However, it differs significantly by application. Bittorrent, un-classified traffic, as well as eDonkey (not shown) experience significantly larger out-of-sequence rates. Bittorrent flows see a mean out-of-sequence rate of more than 5%. In particular, Bittorrent sees excessive mean out-of-sequence rates beyond 8% and sometimes even 10% between 5pm and 8pm. Similar observations hold for the upstream direction (plot not shown). However, for the upstream direction, the high out-of-sequence rates occur during the night and sometimes even exceeds 12%. Note, even during day time, the mean rates do not drop below 4% for Bittorrent and 2% for unclassified traffic.

HTTP traffic on the other hand experiences significantly smaller rates of out-of-sequence packets in both directions. In the downstream direction, HTTP flows see a mean rate of 1.1% out-of-sequence packets, slightly lower than the overall average across all traffic. Even lower are the mean out-of-sequence rates for the classes Well-known and SSL. This is a general observation not specific to the MAR10 dataset. It also holds for AUG08 and AUG09 in both directions.

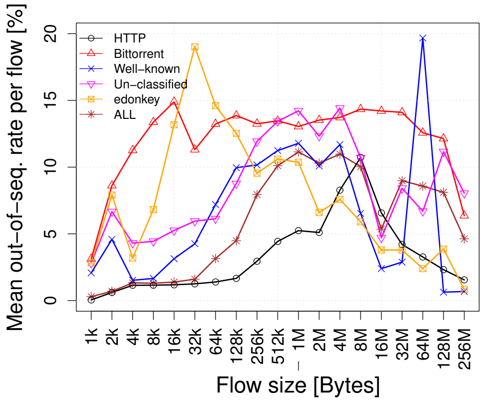
Figures 5.3(b) and 5.3(c) show the overall mean ratio of out-of-sequence packets per application class for the full traces, together with the 95% confidence intervals, for downstream and upstream respectively. These plots confirm that while HTTP, SSL, Well-known, and Other-DPD are experiencing relatively small out-of-sequence rates. Bittorrent, eDonkey, and unclassified traffic see rates that are a factor of 2.5 larger. From Figure 5.3(b) and 5.3(c), we notice a decrease in the total mean out-of-sequence rate across time. However, the mean out-of-sequence rate of Bittorrent does not decrease. Rather, it stays consistently above 5% in the downstream direction and 8% in the upstream direction. Part of the reason is the composition of the traffic: the fraction of Bittorrent and unclassified traffic has decreased while the traffic in the other classes has increased.

The very small out-of-sequence rates for HTTP and well-known, for the DSL upstream traces, stand out. These applications mainly send application requests and TCP ACKs in the upstream direction. Thus, the overall potential of being affected by out-of-order segments is relatively small. It is therefore not surprising to see larger out-of-sequence rates for other applications. Still, the difference between eDonkey and Bittorrent is surprising. Although these are both P2P applications, they seem to impose a different load onto the network, resulting in the different out-of-sequence rates.

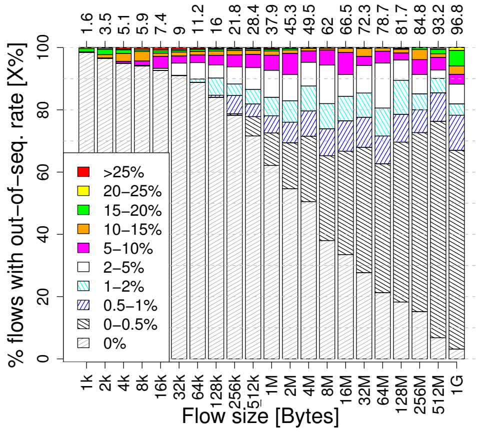
We conclude that **out-of-sequence packets are not evenly distributed across time, nor across application types - P2P and unclassified traffic suffer the most from out-of-sequence packets.**



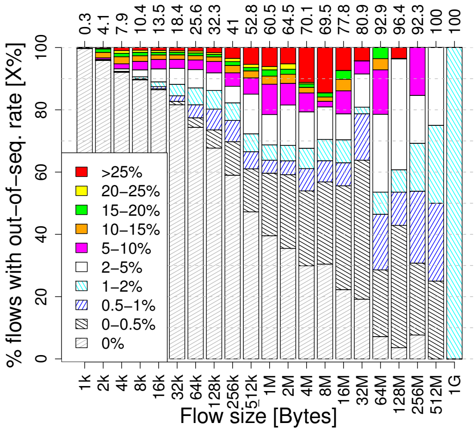
(a) Avg. out-of-seq. rate (MAR10 down).



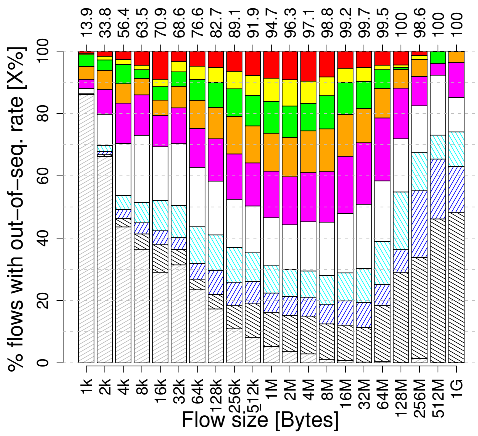
(b) Avg. out-of-seq. rate (MAR10 upstream).



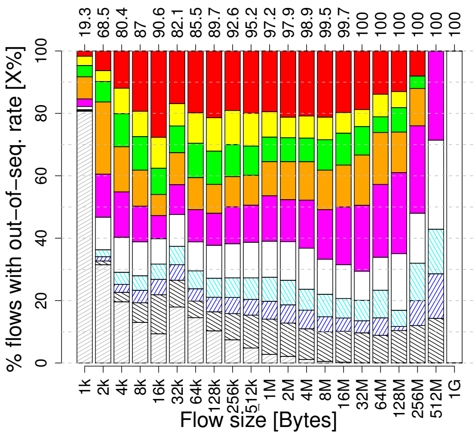
(c) HTTP (MAR10 downstream).



(d) HTTP (MAR10 upstream).



(e) Bittorrent (MAR10 downstream).



(f) Bittorrent (MAR10 upstream).

Figure 5.4: Out-of-sequence packet rate per application for 24h duration. (Fig. (c) and (f) use same legend as Fig. (b))

5.5.2 Flow size—Application type

Since Bittorrent flows experience higher out-of-sequence rates compared to HTTP flows we now take a look at how the out-of-sequence rates for these applications vary across flow sizes. Figures 5.4(a) and 5.4(b) show the mean per-flow out-of-sequence rates across flow-classes, by application type. The upstream and the downstream graphs visually look significantly different for some applications and rather similar for others. Bittorrent, eDonkey, and unclassified dominate the top parts of both graphs with high out-of-sequence rates. HTTP and the other applications are at the bottom for the downstream direction and for some parts of the upstream.

We also see that both for eDonkey as well as for Bittorrent, the out-of-sequence rate first increases with flow size and then decreases. The decrease happens earlier for eDonkey, which is why eDonkey overall sees lower out-of-sequence rates. What happens is that, as the duration of a flow increases, the likelihood of terminating a large flow, that does not perform well enough, increases with the duration of the flow. Large flows should therefore perform better on average than their smaller counterparts. We will show evidence for this in Section 5.5.4. Some of the other spikes in the plots are artifacts of small numbers of flows (less than 100) within a bin, e.g., the spike for Well-Known in the upstream direction. All other bins contain at least 100 flows.

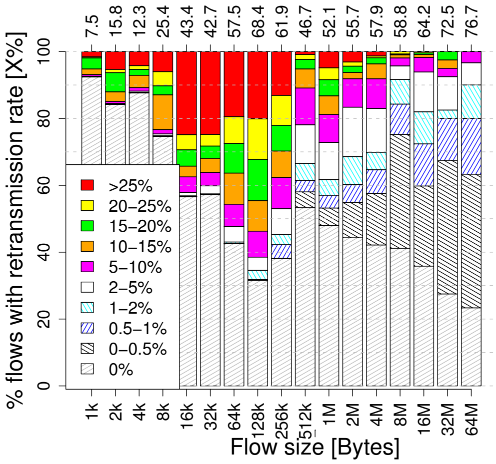
We conclude that **even at the level of specific applications, out-of-sequence packets are not evenly distributed across flow-classes. On average, larger P2P flows experience fewer out-of-sequence packets than medium sized ones. Moreover, there are again significant differences with respect to traffic direction.**

5.5.3 Flow size—HTTP and Bittorrent

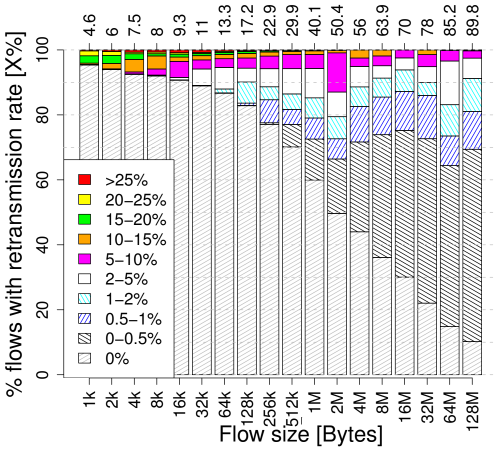
Next, we take a closer look at the ranges of out-of-sequence rates for both HTTP and Bittorrent using the same kind of stacked bar plots as before (e.g., Figure 5.2). Figure 5.4 shows plots for HTTP and Bittorrent for the DSL MAR10 dataset and both directions. Comparing Figure 5.2(a) for the whole trace to Figure 5.4(c) for only HTTP may be puzzling at first—there is hardly any visible difference. However, there are some small difference for short flow-classes and a few larger differences for larger flow classes. HTTP flows experience smaller out-of-sequences rates than one would expect from the results for the overall trace.

Similar observations hold for the upstream direction, i.e., when comparing Figures 5.4(d) for HTTP and 5.2(d) for the full trace. However, this time the differences are larger. Indeed, the best visual agreement of the Figures 5.4(d) for HTTP is with the DSL downstream direction with respect to the fraction of flows that do not experience any or very low out-of-sequence rates.

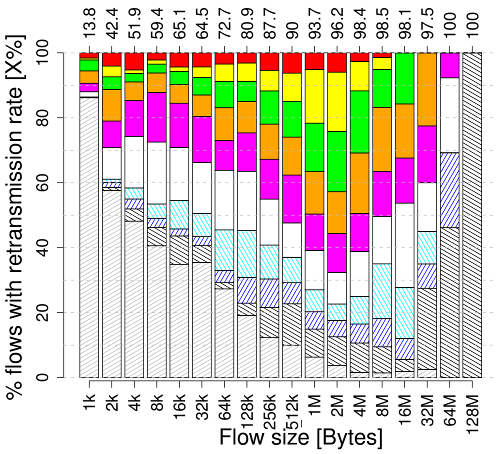
From the comparison of the two upstream directions we find that the out-of-sequence rates are better for HTTP than for the overall—even though worse that for the downstream; a



(a) HTTP (MAWI JP->US).



(b) HTTP (MAR10 downstream).



(c) Bittorrent (MAR10 downstream).

Figure 5.5: Per flow retransmission rate by applications for peak hour. (Fig. (c) uses same legend as Fig (a)).

small percentage of HTTP flows across most flow sizes (except the smallest and largest) experience very high out-of-sequence rates, resulting in the mean out-of-sequence rate from Figure 5.4(b). One of the reason for higher mean out-of-sequence rates are uploads of user generated content: when users upload relatively large (1MB-16MB) files such as large images or short videos, they are restricted by the limited DSL upstream capacity. Other causes can be transfers from servers hosted on the DSL clients networks or multiple parallel transfers, e.g., file sharers.

Turning to Bittorrent we observe from Figure 5.4(e) and 5.4(f) that, as for HTTP in the DSL upstream direction, some Bittorrent flows experience severe out-of-sequence rates, across all flow sizes. In the DSL downstream direction, more than 20% of flows from the classes 4KB-128MB experience out-of-sequence rates larger than 5%. In the DSL downstream direction, this is the case for more than 60% of the flows.

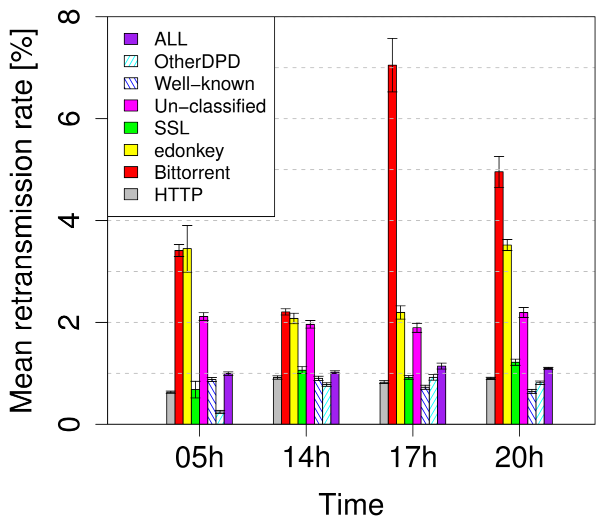
We now return to the comparison of HTTP and Bittorrent. Comparing Figure 5.4(e) with Figure 5.4(c) and Figure 5.4(f) with Figure 5.4(d), we see major differences, both in terms of overall ranges of out-of-sequence rates as well as different distribution across flow-classes. Part of the reason for this larger fraction of out-of-sequence packets is that P2P applications put more demand on both the uplink as well as the downlink. This can cause congestion either on the DSL link, the home network, or the remote network, leading to packet losses which trigger retransmissions. Indeed, visually, the Bittorrent plots for both upstream and downstream are similar to those of the congested JP-US link. In addition, the out-of-sequence rates are in the same order as well, including their distributions across flows.

We conclude that **congestion is one of the main factors responsible for the striking difference in terms of out-of-sequence rates for P2P flows and HTTP flows**.

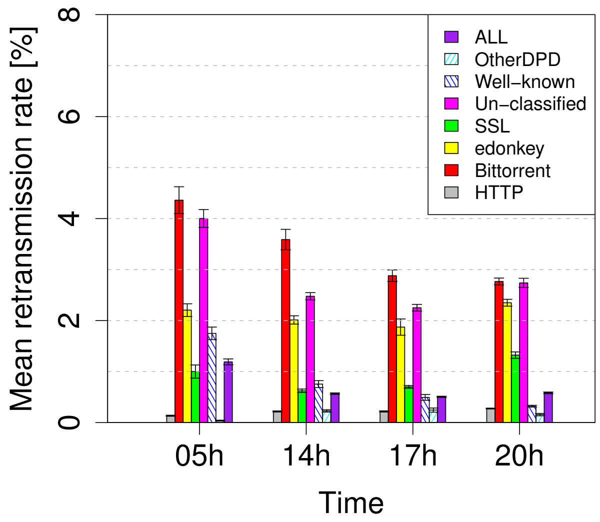
5.5.4 Retransmissions—Across flow size

Next, we focus on retransmissions rather than "only" out-of-sequence packets. Our main motivation for focusing on out-of-sequence packets so far is that these are unambiguous. The identification of retransmissions is more challenging due to (a) the multitude of different OSs and different network stacks and (b) the need to estimate some network parameters. Nevertheless, using the methodology outlined in Section 5.2, we can classify which of the out-of-sequence packets are packet retransmissions. Overall, this fraction depends on the trace but is about 92.6%/85.03% for the DSL downstream/upstream environment and 97% for MAWI for multiple hour-long subsets of the traces. This indicates that out-of-sequence packets are for our purposes a good approximation for retransmitted packets. This agrees with the results of Hurtig et al. [87] who report that about 5.2% of all out-of-sequence packets are due to reordering.

While longer periods such as 24 hours help to illuminate certain aspect for flows which are very long in duration, such as those from P2P applications or applications on well known ports, we next consider shorter time periods to be able to rely on a certain amount of stability. While this may not be the case in general, it is a common assumption, e.g., [114].



(a) RTX (MAR10 downstream).



(b) RTX (MAR10 upstream).

Figure 5.6: Retransmission ratesfor peak hours (MAR10).

Figures 5.5(a) and 5.5(b) show the results for HTTP traffic, again using a stacked barplot across flow-classes. Figure 5.5(a) is for the hour starting at 10pm for the JP to US direction of the MAWI and Figure 5.5(b) for the hour starting at 4:45pm for the downstream direction of the MAR10 dataset. These are the busy hours. Comparing, in particular, the plots for the MAR10 dataset with the one from Figure 5.4(c) we see hardly any difference except that the new plot ends with flow-class 128MB and not 1GB. We do not include the larger flow-classes as the number of samples is very small. Similar observations hold for Figures 5.5(a) and 5.2(b), even though 5.5(a) only includes HTTP traffic. Nevertheless, we see that the general shape is similar. We still observe that some flows have retransmission rates larger than 25% and verified a subset of them manually. Just as before with HTTP we see that a larger fraction of the large flows does not experience any losses. This gives strong evidence that in our traces most of the out-of-sequence packets consist of retransmissions, as was also found by Jaiswal et al. [107] on other data.

Figure 5.5(c) shows the stacked barplot of retransmission rates per flow for only Bittorrent flows in the downstream direction of MAR10. The observation are again consistent with those from Sections 5.5.1–5.5.3. Moreover, the per flow retransmission rates in the upstream direction are again larger (not shown). Returning to Figures 5.5(b) and 5.5(c)—the differences are striking and highlight the widely different ways in which TCP is used by various applications.

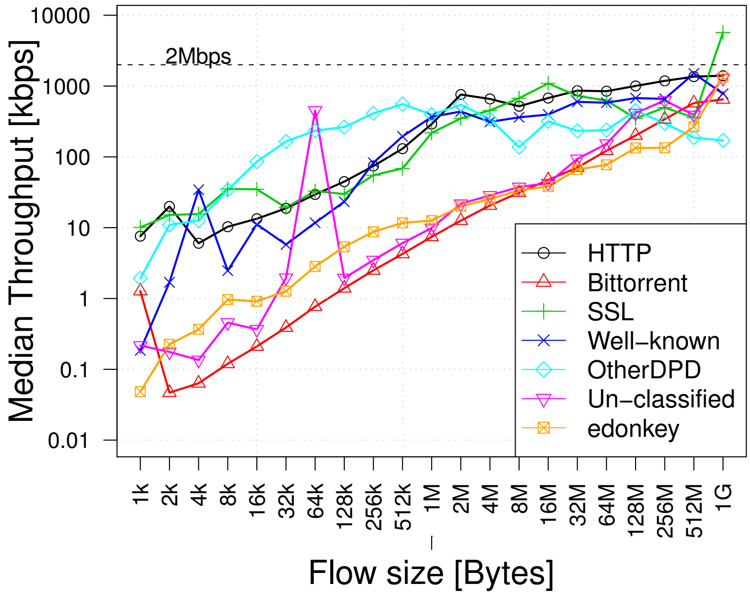
Figure 5.6 shows how the retransmission rate changes across time. These results confirm the results from Figure 5.3(a). During the busy hour the retransmission rate increases in the downstream direction. For the upstream direction, it increases during the off-peak hours. The likely culprit is P2P, as it has a larger fraction of retransmissions relative to the other applications.

We conclude that **the out-of-sequence rate is a reasonable approximation of the retransmission rate. Moreover, even though retransmission rates change over time, P2P and unclassified traffic always tend to dominate in terms relative number of retransmissions.**

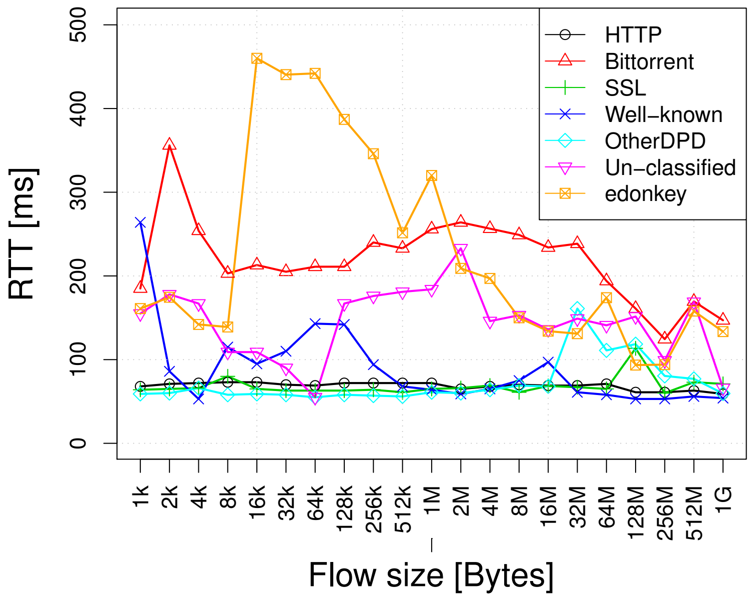
5.5.5 Throughput/RTT—Across flow size

The open question we have not yet answered is if large retransmissions, resp., out-of-sequence rates negatively impact flow performance. In principle, if all TCP mechanisms are well utilized, each retransmission should have a negligible performance impact.

When computing summaries of the flow throughput for each flow class, we find that the performance for flows, e.g., the median, mean, 1st, 3rd quartile, steadily increases from 5Kbps to 250Kbps, 11Kbps to 660Kbps, 0.4Kbps to 68Kbps, and 16Kbps to 740Kbps, respectively, and follows roughly an exponential trend until flow-class 2MB. Then the increase slows down. Part of this is due to the limited DSL capacity. We see a similar behavior for the other network environments, only with different cross over classes and with steeper



(a) Throughput per application.



(b) RTT across different applications.

Figure 5.7: Throughput and RTT(MAR10 downstream).

slopes. We find it surprising that even flows as large as 1MB can expect to get the throughput as larger ones.

Going back to the observation that retransmissions are not distributed evenly, we ask if the same holds for the per flow throughput. Figure 5.7(a) plots the median throughput as received by each application protocol by flow size. Our first observation is that throughput varies drastically across application protocols. Bittorrent, eDonkey, and Un-classified flows experience by far a worse performance than HTTP flows of the same size. This is a first indication that these flows may suffer from retransmissions.

So far the low throughput could still be due to huge RTTs. Figure 5.7(b) thus plots the median RTT per application across flow-classes. We again see notable differences for P2P and HTTP. The RTTs sampled by HTTP do not differ drastically by flow class. This is the contrary for P2P. The latter see increases of the mean RTT well in excess of 200ms. This indicates that either the network distances are significantly larger for P2P traffic than HTTP, or there are significant queues in the network and that the queues are contributing to the delay. Verification of the geolocation of the addresses shows that the P2P end-points are a bit further away but not in such a way as to justify an increase in the median RTT from roughly 60ms to more than 200ms. Thus, we conclude that buffering is partially responsible for the RTT increase.

Another observation from Figure 5.7 is that the difference between the applications—in throughput and in RTT—decreases as we consider larger and larger flow sizes. This indicates that the large P2P flows receive reasonable performance which is consistent with our observation that their retransmission rates are lower. We presume that there is some amount of self-selection. Almost all P2P protocols include a mechanism to prefer well performing peers over those that are not, ensuring reasonable performance for the large flows.

We conclude that **neither throughput nor RTTs are distributed evenly across flow-classes or application types. Moreover, as the flow size increases, the performance of P2P is approaching that of HTTP (measured in terms of throughput and RTT).**

5.5.6 Summary

From examining traces from three different network locations we learn that:

- The out-of-sequence rate is a reasonable approximation of the retransmission rate.
- Retransmissions are not evenly distributed across time nor application types—P2P and unclassified traffic dominates—nor flow sizes—medium flow-classes dominate.
- Congestion is responsible for the striking difference in terms of out-of-sequence rates for P2P flows and HTTP flows. This congestion can be locally induced by too many parallel connections, remotely induced by too many connections to the remote host, or network-induced by insufficient bandwidth.

- Neither throughput nor round-trip-times are distributed evenly across flow sizes or application types. Moreover, as the flow size increases the performance of P2P is getting closer to that of HTTP.

These insights are succinctly summarized as Lesson 2 in Section 5.1.

5.6 Local/remote retransmissions

Next, we ask the question which network segment is responsible for all those retransmissions. For this purpose, we rely on the methodology from Section 5.2 to separate retransmissions from before and after the monitoring point.

Figure 5.8 shows the resulting retransmission rates across flow classes separated according to their location for both downstream and upstream. We again see the striking difference between All, HTTP, and Bittorrent. These result relate very well to those from Section 5.5.

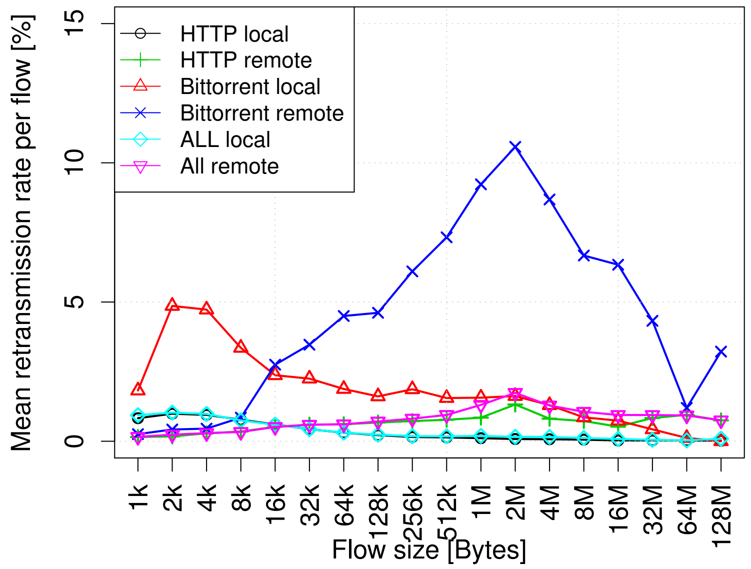
With respect to downstream, we find that the cause for most of the retransmissions for all applications is the remote segment. This is in particular striking for Bittorrent. Some possible explanations for this includes the limited upstream capacity of other peers and/or cross traffic. In addition, the local segment is the culprit for most retransmissions of short flows, in particular for those from Bittorrent. A possible explanation is the ever increasing complexity of the customers' home networks with various numbers of wireless devices.

With respect to upstream flows, we observe that most flows have already been impacted before they even reach the monitor. The big spike for medium sized flows greater than 5% in Figure 5.8(b) confirms this. We also observe that small flows in the upstream direction are mainly impacted by the remote side, caused by conditions that these flows encountered before reaching the other end-host. Overall, this corresponds to an almost symmetric scenario.

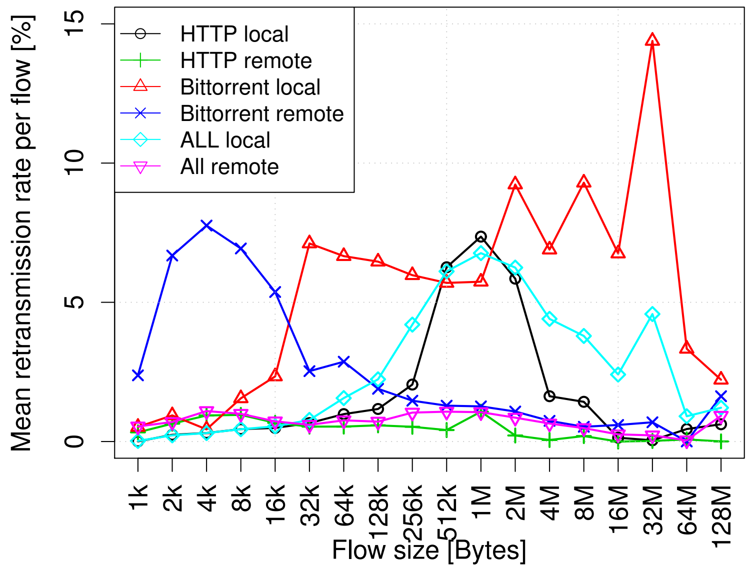
We conclude that **irrespective of the considered applications, while most retransmissions in the upstream direction are triggered by events within the network segment between the DSL customer and the monitoring point, most retransmissions in the downstream direction are caused by events upstream from the monitoring point (i.e., rest of the Internet.** These observations are captured by Lesson 3 in Section 5.1.

5.7 Timeouts or fast recovery?

Finally, we ask which TCP loss recovery mechanism is used by flows and does it differ across flow-classes. TCP loss recovery is one of the essential components of TCP. However, there are many variants. Some recovery mechanisms are common to all TCP flavors, including retransmission time-outs (RTO) as well as fast retransmissions (FRTX). They



(a) RTX (MAR10 downstream).



(b) RTX (MAR10 upstream).

Figure 5.8: Local, remote, and total retransmission rates for peak hour.

are foundations of TCP loss recovery. When a time-out for a segment occurs, TCP retransmits the segment (RTO) and continues recovery for the subsequent segments in the flight (RTOREC). After receiving three duplicate ACKs, TCP initiates fast retransmission (FRTX) of the lost segment and continues recovery for the remaining flight (FREC). Note, fast retransmissions (FREC) also include those retransmitted with the help of SACK.

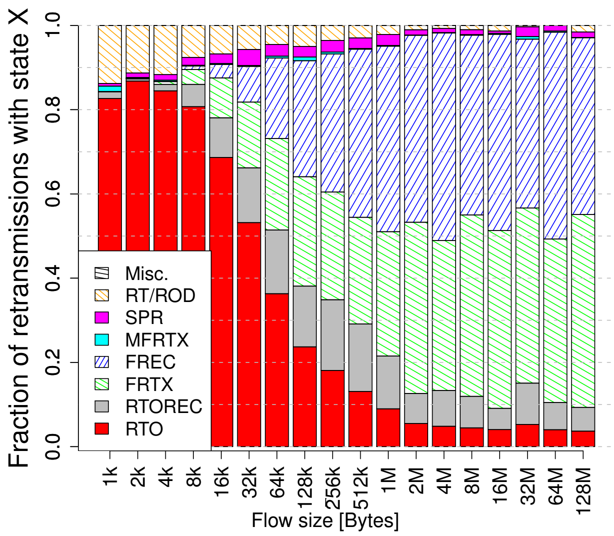
Some TCP implementations use refined mechanism. For example, Microsoft's TCP often triggers fast retransmissions after only two duplicate ACKs (MFRTX) rather than the standard three (FRTX). In some other situations, TCP senders send retransmissions that are unnecessary. These are referred to as spurious retransmissions (SPR) which can even get re-ordered (RT/ROD). All other retransmissions are summarized under miscellaneous (Misc).

We rely on this retransmission classification to study how TCP recovers from retransmissions for flows of different sizes and from different applications. However, before we consider the results, let us step back and reflect on what to expect. Short flows, as they consist of only a few packets, should not be able to recover from losses using fast recovery (F/FR) which can be used by large flows. However, what is the minimum flow size from which TCP is able to rely on fast recovery in practice. Recall, TCP estimates RTO values based on the current round-trip times estimations and their variance. A typical initial value is 3 seconds and is updated to be the estimated mean RTT plus four times the estimated variance. Moreover, the TCP specification imposes a 1 second minimum RTO [156]. Thus, any RTO event has a significant impact on the flow performance [156].

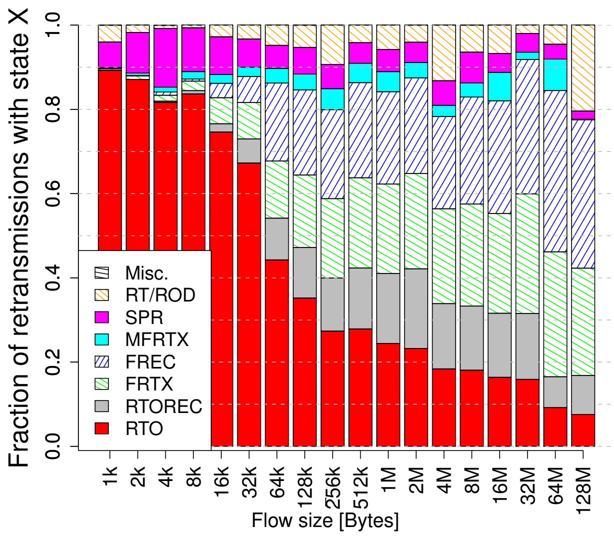
Figure 5.9 shows, as stacked barplot, for each flow size which TCP loss recovery mechanism has been used for each retransmitted packet, as classified by tcpcsm. Figures 5.9(a) resp. 5.9(b) show the results for HTTP/Bittorrent for the DSL downstream direction of MAR10. We observe that the majority of flows smaller than 64KB have their retransmissions triggered by retransmission time-outs (RTO) or recovery events immediately following RTO (RTOREC). Flows smaller than 1MB see a large fraction of time-out related events, which affects their performance negatively. Similar observations hold for the upstream and the other traces and environments.

Figure 5.9 explains the better throughputs for flows larger than 1MB observed in Section 5.5: most of the retransmissions benefit from fast recovery (FRTX and FREC). Indeed, we see that no more than 15% of the retransmissions are due to timeouts for flows larger than 1MB. The main difference between HTTP and Bittorrent is in the significant fraction of spurious retransmissions (SPR) or re-ordered retransmissions (RT/ROD) for Bittorrent flows of all flow-classes. Such retransmissions can impede flow throughput significantly and we likely observed the results in Section 5.5. Large HTTP flows on the other hand see few if any of these spurious or re-ordered retransmissions.

We conclude that **since short flows typically rely on timeouts for loss recovery, their performance is negatively affected whenever they experience losses. Furthermore, one of the reasons for the low throughput of P2P flows is TCP loss recovery which in this case leads to a significant fraction of spurious retransmissions. On the other hand,**



(a) HTTP



(b) Bittorrent

Figure 5.9: TCP recovery mechanisms for peak hour (MAR10 downstream).

large HTTP flows do not suffer from this problem. These observations correspond to Lesson 4 in Section 5.1.

5.8 Related work

Much work in the past has tried to better understand Internet properties, for example packet loss [52, 75, 145]. Akella et al. [37] studied Internet bottlenecks, and found that they occur equally within ISPs as well as across peering links. Aikat et al. [35] studied the variability of TCP round-trip times within a connection, and found that the RTT values vary widely. But et al. [55] presented an algorithm to estimate running RTT and jitter characteristics of TCP streams monitored at the midpoint of a TCP flow.

Lar et al. [118] provide a very comprehensive review of TCP congestion control mechanisms. Siekkinen et al. [181] proposed a TCP toolkit, able to find the primary cause of throughput limitations of TCP flows. Mellia et al. [139] proposed a tool, tstat, that computes detailed performance statistics at both the IP and TCP layers.

Wang et al. [196] studied packet reordering in the Internet, and proposed a approach to infer reorder-generating spots in the Internet. Mellia et al. [140] proposed a new heuristic technique to classify TCP anomalies, including out-of-sequence and duplicate segments.

Qian et al. [165], in particular, exposed the prevalence of irregular retransmissions across different flow sizes in the Internet. Hurtig et al. [87] have also reported that packet reordering has reduced and is in the order of 5.2% of all out-of-sequence packets. Zhang et al. [206] have found that flow size and flow rate are two highly correlated metrics. The relationship between short flows and application performance has also been studied by Hafsaoui et al. [83]. Recently, Dukkipati et al. [70] has proposed to increase the initial congestion window to 10 to save round trip times for better response times. Similarly, another work from Dukkipati et al. [69] has found that the fast recovery mechanism behaves in a bursty manner and fast recovery should be done using a proportional rate. The behavior of large flows spanning days is discussed in Quan et al. [166]. Similarly, Lee et al. [119] have studied the performance of a congested academic link but not focused on the short flow performance. They found mean loss rates of 5.77%, which are consistent with our results. A number of papers focused on the performance of TCP in wireless networks. For example, Gerber et al. [77] focus is on estimating throughput of TCP flows for 3G wireless networks. Vacirca et al. [191] studied the presence of spurious TCP retransmissions in UMTS/GPRS networks.

In the context of applications and losses, Alcock et al. [38] have exposed the problem of YouTube block send, that causes unexpected losses. Izal et al. [103] have studied the behavior and performance of Bittorrent over a period of multiple months. Pouwelse et al. [162] studied multiple performance aspects of the Bittorrent protocol.

5.9 Summary

We study flow-level performance of popular applications across flow sizes. Our metrics to gauge flow performance include retransmissions, out-of-sequence packets, throughput, and round-trip times. Based on traces gathered from three very different network environments, we compare flow performance through the lens of these metrics, under different network loads, access link capacities, and traffic directions.

We find that network load has a significant impact on flow performance, and that different applications are impacted differently. We also find that flow performance varies significantly across flow sizes. For example, contrary to popular belief, small flows, that make up a majority of the flows, experience significant retransmission rates, across all applications. Large flows on the other hand, although fewer in number, can experience limited retransmissions. We observe a marked contrast between HTTP and P2P flows. Indeed, P2P flows suffer from continuously high retransmissions compared to HTTP. We identify the access part of the network as the area of the network responsible for these retransmissions, not the network core. Moreover, part of the bad performance of P2P is related to spurious retransmissions.

6

VoIP QoE prediction in NGMN

6.1 Overview

The disparity in flow performance we find in the traces is highlighting QoS variations for the flows that represent a large Internet user-base. Predicting user experience from the information available from the middle of the network is still an active area of research. Our observations based on the information collected from the middle of the network, in quality engineering sense, do not represent the *real* user experience. However, our observations lead us to craft specific user experience studies for VoIP and video applications which users deem critical while using the Internet. For this purpose, we confine our study to NGMN environment. NGMNs promise an all-IP platform for a seamless user experience for the current and future Internet. We select several different combinations of frequently occurring networking conditions in NGMN environment to investigate the quality dimension. We thus embark in the next three chapters on exploring user experience in NGMNs for VoIP and video applications.

As discussed earlier, over the last decade three trends have emerged: a) VoIP services have become very popular, e.g., Skype, b) sophisticated mobile devices are available, e.g., the iPhone, c) high speed wireless networks are available almost everywhere in the developed world, e.g., UMTS, WiFi, WiMAX. Motivated by these trends, we try to understand the impact of Next Generation Mobile Networks (NGMN) on VoIP service as such technologies cause time-varying degradations that affect user perception. In particular, we concentrate on mobile users. Actions that may affect the VoIP quality of mobile users include *network handover*, i.e., switching from one wireless network to another while changing the network attachment point, *codec changeover*, i.e., switching from a narrowband to a wideband codec, and *transmission degradations*, i.e., packet loss or discard in the network.

Operators need to monitor the quality of the service that their users receive. Call quality monitoring can be based either on network metrics, an evaluation of the received audio signal, or on user feedback. The first represent quantitative metrics of performance and as such measure the Quality of Service (QoS). The latter two address the Quality of Experience (QoE) as experienced by the user. However, the evaluation of the signal can only yield an estimation via a perception model of the users QoE. A users QoE can be measured via auditory tests, e.g., according to ITU-T Rec. P.800 [99], which results in a Mean Opinion Score (MOS). However, this is not only time-consuming and expensive but also not applicable for online monitoring. Therefore, one commonly uses perception models which rely on instrumental quality prediction.

Typical examples of such instrumental quality prediction models include the Perceptual Evaluation of Speech Quality – PESQ model [100], which relies on the analysis of the signal waveform received by the end-system, and the E-model [92], which relies on an evaluation of the network and terminal properties. In this chapter, we focus on the wideband PESQ (WB-PESQ) model [101] as it should in principle be capable of capturing the time-varying degradations entailed by NGMN given that it is based on the signal received by the terminal. In the remainder of the chapter we use WB-PESQ.

We note that current quality prediction models have been developed based on a limited set of scenarios. A typical scenario includes a specific setup of the terminals, the network, as well as the transmission conditions. However, to the best of our knowledge, none of the signal-based models have been thoroughly validated for scenarios that consider the impact of NGMN, where network handovers are common, codecs may suddenly be switched, or transmission metrics may differ drastically between networks. Therefore, in this chapter, we endeavor to bridge this gap by a comprehensive evaluation of WB-PESQ quality estimations in NGMNs for the following set of scenarios: 1) wideband-narrowband speech codec switching, 2) speech signal fading during codec switching, 3) and talk-spurt internal time-shifting due to jitter buffer size differences. We find that WB-PESQ fails to correctly predict the quality degradations entailed by these NGMN effects. By pointing out the impact of these degradations on the speech signal, this chapter contributes to potential improvements and adaptation of the wideband PESQ model for NGMNs.

We organize the chapter as follows. Section 6.2 describes our methodology and approach employed to perform the experiments. Section 6.3 presents an evaluation of WB-PESQ prediction in NGMNs via a statistical analysis. Section 6.4 discusses related previous work on user perception models. The chapter concludes with Section 6.5 that summarizes our main findings.

6.2 Methodology

To evaluate the quality of WB-PESQ predictions under NGMN effects, we design a set of controlled experiments within the QoE-Lab testbed framework and then perform auditory quality tests, as well as a detailed analysis of the resulting signal and network effects.

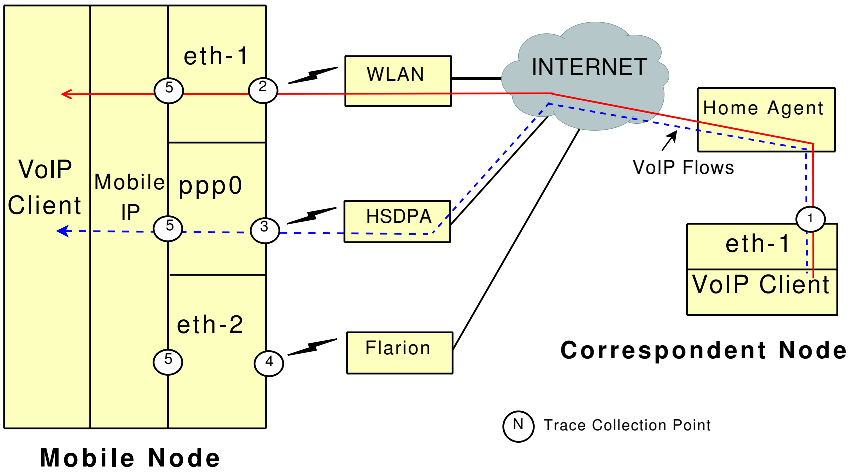


Figure 6.1: QoE-Lab experimental setup for VoIP services in NGMN

6.2.1 QoE-Lab NGMN testbed

QoE-Lab offers mobile clients the capabilities to perform network handovers and speech codecs changes even during ongoing VoIP calls. In addition, the testbed includes an extensive measurement infrastructure that enables us to simultaneously capture network traces at all network interfaces and the audio streams at the terminals. The testbed includes the following access networks: 1) WiFi/WLAN, 2) Universal Mobile Telecommunications System (UMTS) / High-Speed Downlink Packet Access (HSDPA) and 3) Flarion – Flash-OFDM (pre-WiMax release).

We rely on Mobile IP, more specifically on the *SecGo 3.2* [20] Mobile IP toolkit, to enable seamless handover between different radio access technologies. We use Mobile IPv4 since not all access networks support IPv6 at the time of the experiments.

We choose the open source VoIP framework *PJPROJECT 0.5.10.3* [18] based on SIP and rely on the seamless codec switching solution developed and evaluated by Wältermann et al. [194]. It is based on SIP/SDP parameter renegotiation [171], and utilizes parallel media stream establishment with internal RTP packet routing in the application. This approach reduces the potential of application-layer packet losses due to the codec changeovers. For wideband transmission the ITU-T G.722.2 (AMR-WB) codec at 23.05 kb/s and for narrowband (NB) the ITU-T G.711 log. PCM codec at 64 kb/s are used with their native packet loss concealment (PLC), respectively. In addition, *PJPROJECT* has been instrumented to obtain jitter buffer logs.

6.2.1.1 Hardware and Software Components

The main elements of the QoE-Lab testbed for NGMN (Section 3.2.2), as shown in Figure 6.1, are the Mobile Node (MN), the Correspondent Node (CN), and the Home Agent (HA). The MN can utilize either of the following access technologies: WLAN, HSDPA and Flarion. The CN and MN are laptops running Linux 2.6.18.2. The HA is realized using a *Cisco 2620XM* router. All VoIP calls are originated at the CN, communicate via the Internet, and are terminated at the MN to capture mobile user perception. During a call, the VoIP path is determined by selecting the appropriate network interface, see the solid and dashed lines in Figure 6.1. Each numbered circle shows a network trace collection point. To realize transmission degradation we use *netem* at the mobile node.

6.2.2 Approach

To be able to test the WB-PESQ quality estimation capabilities for NGMNs under network handovers, codec changeovers, and packet-loss degradations, we designed 26 scenarios/-conditions. With regards to network handover we consider switching between WLAN, HSDPA, and Flarion networks. For codec changeover we consider the wideband codec ITU-T G.722.2 and the narrowband codec ITU-T G.711. With regards to transmission degradation we introduce random packet losses (P_{pl}) at various rates ranging from none to 20%. Given that it was not possible to realize a full combination of all scenarios due to the limited scalability of subjective tests we focus on the subset that includes all variants of network switching and codec switching (in both directions). We also added some scenarios with transmission impairments. Moreover, to eliminate speaker bias, 10 different speech samples (originating from two male and two female speakers) were picked. All in all, this results in a balanced set of speech samples which consist of an approximately equal number of good and bad VoIP quality samples.

Based on speech samples collected from the Mobisense testbed under controlled experimental conditions, we performed both auditory quality tests and an analysis of the network effects. On the perception plane the following steps have been performed: for each NGMN-condition 1) the audio samples were generated, 2) these were judged in an auditory experiment to obtain average quality scores (MOSs), and 3) analyzed using the WB-PESQ model. On the networking plane, 1) network traces were collected on the transmission endpoints and 2) used for the network trace analysis. Then, the results of both planes are merged to evaluate WB-PESQ performance in NGMN conditions focusing on codec changeover, network handover, and packet loss variations. As such most experiments have three phases: pre-switching, switching, and post-switching, where the switch can be any combination of codec changeover, network handover, or degree of packet loss rates.

In general, auditory quality tests are capable of evaluating how users perceive the quality of an entire call as well as of short speech samples. However, since signal-based quality prediction methods are unable to predict quality for the overall call, in this chapter we focus only on the assessment of short samples using auditory test according to ITU-T Rec. P.800 [99].

Our balanced set of speech samples that covers the quality range from "Excellent" to "Bad" has been presented to a set of 24 test participants in a sound insulated laboratory. For the quality rating, a 5-point Absolute Category Rating (ACR) scale was used, resulting in averaged Mean Opinion Scores (MOSs) for each test condition, see [142] for details. The WB-PESQ predictions were obtained for all the samples presented to the listeners.

6.2.3 Trace analysis

Utilizing the network traces gathered using *tcpdump* at the network interfaces of CN and MN (points 1 to 5 in Figure 6.1) we are able to calculate network statistics including inter-packet delay (IPD), jitter, and packet-loss. Given that these metrics do differ substantially during the three phases, we calculate them separately for each phase. For this purpose we also use the collected data to identify when the codec changeover or the network handover occurred. The former is possible by identifying the different RTP sequence series and Payload Type in the traces at the MN. The latter is more complicated and is done by merging the various packet traces and observing when the actual switch happened as the VoIP stream is partially captured at each of the physical access interfaces of the MN. This enables us to separate the impact of the specific speech codec and the specific network transmission technologies on the VoIP stream.

We augment our network trace analysis with a jitter buffer log analysis which we obtain from the VoIP application. This provides us with the overall packet loss statistics which the VoIP decoder experiences. It allows us to contrast the speech signal degradations with the degradations due to network effects.

6.3 Evaluation of WB-PESQ in NGMN

In this Section we start with quantifying the differences between the auditory MOS and the predictions of the WB-PESQ model. To ascertain the main contributors to the discrepancies, we perform a *sensitivity analysis* and then try to determine which NGMN effect is responsible.

6.3.1 User MOS vs. WB-PESQ estimation

Table 6.1 summarized for each experiment the user quality judgements (MOS), WB-PESQ quality estimation and their difference (Diff). We find that overall the WB-PESQ-model quality prediction is close to the user opinion resulting in the Pearson correlation $r = 0.956$ and a root mean squared error (RMSE) $\sigma = 0.471$.

However, we also find some rather significant differences between the prediction and the user MOS. WB-PESQ consistently underestimates the user experience for certain scenarios/conditions with NGMN effects, see Table 6.1 which is ordered by their difference.

Table 6.1: Comparison of WB-PESQ quality estimates and Auditory Judgments (MOS). Ppl: Packet loss in %; Diff.: Difference of MOS & WB-PESQ; W: WLAN; H: HSDPA; F: Flarion; →: Handover/Changeover, a./b. : Changeover before/after handover.

No.	Network	Codec	Ppl	MOS	WB-PESQ	Diff.
1	H	711→722.2	0	2.33	1.47	0.86
2	H→W(b.)	711→722.2	0	2.59	1.81	0.78
3	W→H	722.2	0	4.27	3.49	0.78
4	W	722.2	0	4.49	3.74	0.75
5	F→H(b.)	722.2→711	0	2.54	1.84	0.70
6	F→H(a.)	722.2→711	0	2.48	1.81	0.67
7	H→F(b.)	711→722.2	0	2.28	1.65	0.63
8	W→H(b.)	722.2→711	0	2.34	1.72	0.62
9	F→H	722.2	0	4.06	3.44	0.62
10	H→W(a.)	711→722.2	0	2.55	1.95	0.6
11	H	722.2→711	0	2.29	1.70	0.59
12	H→F(a.)	711→722.2	0	2.57	1.99	0.58
13	H→F	711	0	3.11	2.66	0.45
14	H→F	711	10	2.43	2.04	0.39
15	H→W	711	10	2.42	2.03	0.39
16	W	722.2	10	2.02	1.65	0.37
17	H→W(b.)	711→722.2	20	1.81	1.48	0.33
18	W→H(a.)	722.2→711	0	2.32	2.01	0.31
19	W→H	722.2	10	2.34	2.06	0.28
20	H	711	0	2.95	2.68	0.27
21	H→W	711	0	3.28	3.02	0.26
22	F→H	722.2	10	2.16	1.94	0.22
23	H	711	10	1.96	1.77	0.19
24	W→H(b.)	722.2→711	20	1.38	1.24	0.14
25	W	722.2	20	1.27	1.26	0.01
26	H	711	20	1.45	1.48	-0.03

Table 6.2: ANOVA - percentage of variation explained by individual factors("%") and statistical significance (F-value, "F").

	Network		Codec		Packet loss		Res.
	%	F	%	F	%	F	%
MOS	2.9	3.8	12.5	16.6	54.7	217.0	16.1
WB-PESQ	2.4	2.8	22.8	26.3	42.2	146.3	18.4
Difference	5.2	2.1	21.4	10.1	13.1	16.2	51.7

The largest WB-PESQ-model inaccuracy of 0.86 MOS points is observed when switching codec from narrowband to wideband (condition #1). Minor deviations are observed in setups without codec switching (conditions #25,26) and without codec and network switching (#23,20,16).

In principle, we note that the user MOS is good (> 4) as long as a wideband codec is used. Yet, the WB-PESQ estimation is significantly lower. While switching codec lowers the user MOS below 2.6, switching networks does not necessarily lower the user MOS in the same manner, see for example conditions #3 and #9. The lowest user MOS values are for conditions with large packet losses (conditions #17, #24–26). The WB-PESQ estimations for these conditions are rather close which indicates that WB-PESQ is very much capable of capturing the effects of packet losses. Moreover, for conditions without packet losses, e.g., #1–13, the match can be off by up to an absolute value of 0.86 or 36%. We note that this set of conditions includes some with only codec changeovers in both directions as well as some with a variety of network handovers. In addition, some include both effects. Condition #11 points out that a mere codec change from wideband to narrowband already substantially impairs user MOS; the WB-PESQ estimate is even more drastic in this case.

We also note that as the packet loss rate is increased, the differences between user MOS and WB-PESQ estimate decrease. This indicates that the effects of the packet loss can mask the effects introduced by NGMN handovers and changeovers. However, this could also be an artifact related to scale (1–5) of the MOS scores, which saturates close to the MOS-scale minimum of 1 for very bad quality. The above discussion summarizes some fairly obvious conclusion that can be drawn from Table 6.1. Yet, to understand their statistical significance we perform an *Analysis of Variance* (ANOVA) [106] on the experiments.

6.3.2 Factor analysis

The first step in an *ANOVA* analysis is to identify the factors, their grouping, and their levels. In our experiments, we have the following 3 factors: network condition, codec, and packet loss; with the levels: network conditions (WLAN, HSDPA, WLAN→HSDPA, HSDPA→WLAN), codec (NB, WB, NB→WB, WB→NB), packet loss (0-1, 1-5, 10-20,

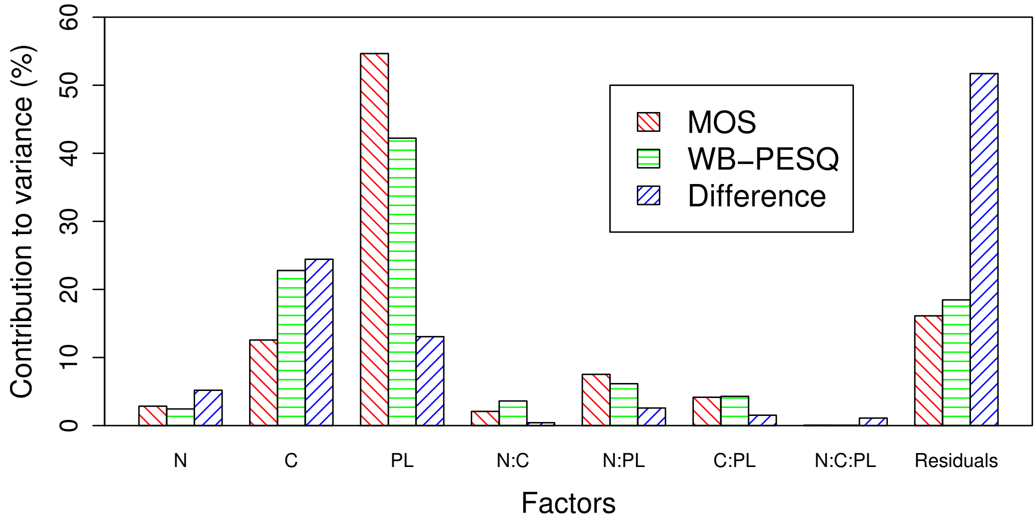


Figure 6.2: Relative contribution of individual factors to total variation (sum of squares); N: Network, C: Codec, PL: Packet-loss

20+)%. Even though some of the combinations of levels have not been tested, the experimental data is rich enough to determine the most influential factors for the auditory MOS, the WB-PESQ estimations, and their differences. In addition, we use a regression model to capture factor interactions. Moreover, to study if the differences in the scores are caused by the factors or by the noise (due to running multiple experiments for each factor combination) we partition the observed variance of each metric into their components using the *sum of squares* and use F-tests to identify the statistically significant factors.

The results are summarized in Table 6.2 and visualized in Figure 6.2. Rather than showing the variation that a certain factor contributes to the total variance of each metric we show its relative contribution. Hence, a high value for a factor suggest that the score is very sensitive to that factor. We see that the most influential factor for MOS and WB-PESQ is packet loss.

We note that packet loss is the highest contributor for MOS (54.7%) and WB-PESQ (42.2%) scores which confirms our previous observations that any significant packet loss shapes the perceived quality. Codec is the second largest contributer to MOS (12.5%) and WB-PESQ (22.8%) while network effects are significantly smaller. Most interestingly, we find that the reported differences between MOS and WB-PESQ are mainly due to codec effects (21.4%) followed by packet loss (13.1%). However, this analysis also yields a large residual. Nevertheless, it points out that a closer analysis of the effects of codec switching is needed. For this purpose we investigate particular speech samples under NGMN conditions next.

6.3.3 Critical conditions in NGMN

To understand why WB-PESQ estimates are more critical than the user judgements we have to answer the following questions: 1) How is the output signal degraded by codec changeover and/or network handover? 2) Does such a signal degradation affect user perception? 3) How does WB-PESQ treat such impairments?

Based upon a detailed analysis of individual audio signals we identify three main types of audio signal degradations:

Effect 1 – Internal talk-spurt shifting due to network link problems:

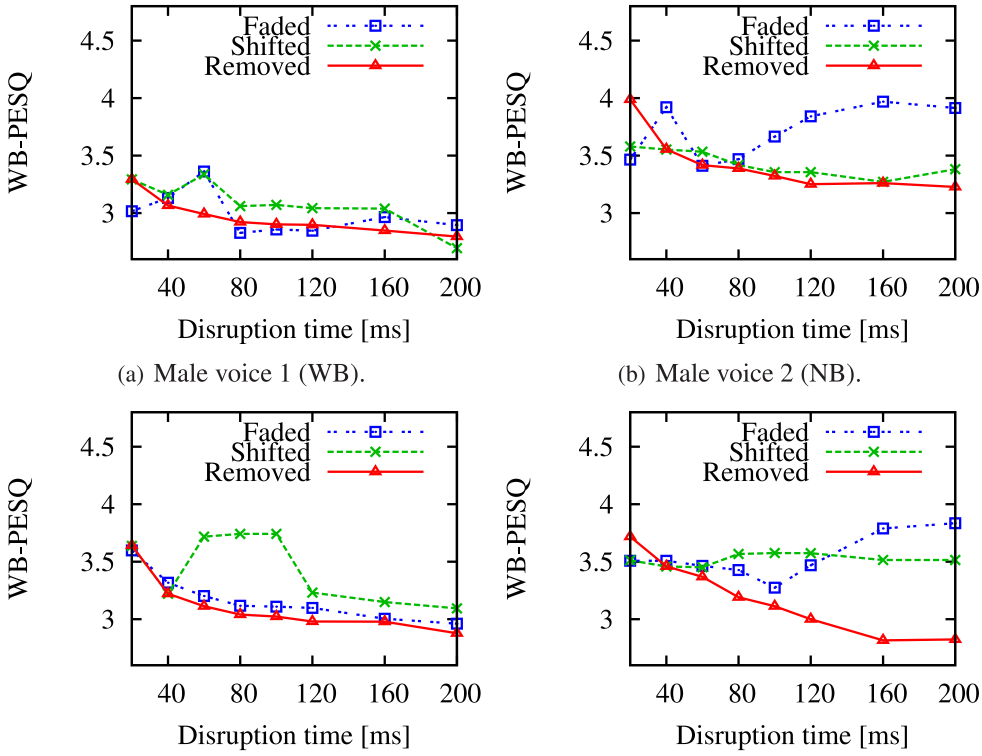
Certain wireless network technologies suffer from time-varying properties that can introduce large jitter in the packet inter-arrival times, linked for example to a network handover or to networks with varying transmission link qualities, e.g., UMTS/HSDPA in 3G or LTE (Long Term Evolution) in 4G. This does not only influence the adaptive buffer size but it can also cause the VoIP jitter buffer to underflow which in turn results in audio signal pauses albeit no data is lost. We point out that WB-PESQ tries to time-align the reference and degraded signals globally and locally: WB-PESQ first aligns the utterances (sentences), and if the delay due to time-varying conditions changes within one utterance, it aligns the two sub-utterances. However, sub-utterances must be of at least 800ms. Shorter sub-utterance are not aligned, and subsequently WB-PESQ does not try to re-align the main utterance. Also, if the delay appears between two utterances, WB-PESQ does not re-align the reference signal. Hence WB-PESQ partially fails to cope with the time-shifting of internal talkspurts due to time-varying conditions in NGMN. While this causes a degradation in the WB-PESQ score, this does not appear to impact user perception to a noticeable degree.

Effect 2 – Audio signal fading during codec changeover:

Due to the design of the codec changeover algorithm, the application may fade the speech signal during the codec switching period. The duration of this audio signal disruption is related to the state and size of the individual VoIP buffers, and it is small if the size of both buffers is roughly the same. However, buffer sizes are determined adaptively. Therefore, their sizes can differ and depend on the network characteristics. WB-PESQ provides a global normalization for loudness over the whole speech signal and a partial time-varying gain compensation. If this short-term gain is above +10dB or below -10dB, the gain is not fully compensated. The gain compensation could even influence the frequency compensation and introduce noise in the compensated reference signal. This effect does not have a strong impact on the user perception but is punished by WB-PESQ.

Effect 3 - Speech bandwidth change:

Wideband-narrowband codec changeovers can happen in NGMNs. However, existing quality prediction models are designed for predicting one-bandwidth speech quality only. Therefore, internal algorithm routines like the frequency response calculation and bandpass filter banks do not consider this kind of phenomena yet. This effect has a significant impact on user perception but is not properly handled by WB-PESQ as supported by our findings.



(a) Male voice 1 (WB).

(b) Male voice 2 (NB).

(c) Female voice 1 (WB).

(d) Female voice 2 (NB).

Figure 6.3: WB-PESQ quality estimation for signals malformed by emulated NGMN degradations: i) Signal fading (Faded), ii) Signal time-shifting (Shifted), and iii) Signal loss (Removed).

To evaluate the impact of the NGMN effects 1 and 2 on the difference between the WB-PESQ prediction and the user perception we designed additional experiments. We selected four speech samples – two female voices, one in wideband and one in narrowband, and two male voices – and degraded them in a controlled manner. For the degradation we pick a disruption time in the middle of the speech sample, simulate signal fading and signal time shift. As a comparison we also simulate packet loss by replacing signal parts with silence periods. The duration of the disruption time ranges from 20-200ms.

Figure 6.3 shows the resulting WB-PESQ scores for four different speech samples. Our hypothesis is that the WB-PESQ scores should decrease as the disruption time is increased. This hypothesis is true for our comparison effect *Signal loss* for both WB & NB and *Signal Fading* for WB only. However, it does not hold completely for the *Signal time-shifting* effects. The quality estimation is not monotonically decreasing and its shape differs substantially for this condition. In these cases, the WB-PESQ model provides counter-intuitive results. The potential cause of this variation is related to the question which part of the speech signal is impaired, as their importance can vary.

In summary, our results question the applicability of WB-PESQ estimations for specific NGMN conditions and the WB-PESQ model should be improved to handle NGMN-style degradations in a more reliable manner. the WB-PESQ prediction and the user perception we designed additional experiments. We selected four speech samples – two female voices, one in wideband and one in narrowband, and two male voices – and degraded them in a controlled manner. For the degradation we pick a disruption time in the middle of the speech sample, simulate signal fading and signal time shift. As a comparison we also simulate packet loss by replacing signal parts with silence periods. The duration of the disruption time ranges from 20-200ms.

6.4 Related work

Currently network operators are in the process of upgrading their fixed as well as mobile network infrastructure to support wideband voice services. This network upgrade has lead to an improvement of the speech quality perceived by the users by about 30% [141, 167]. However, the impact of time-varying degradations inflicted by the usage of NGMN is still not well understood [193].

To predict the quality of wideband speech transmission the quality prediction models have been extended, e.g., ITU-T G.107 Amendment 1 [91] is an extension of the parameter-based E-model [92] and ITU-T P.862.2 [101], referred to as wideband PESQ (WB-PESQ) is an extension of the PESQ algorithm [100] that has been specifically optimized to handle wideband signal distortions.

Initial work on integrating impairments imposed by NGMN effects to the E-model, as well as the PESQ model have been done. For instance, Lewcio et al. [125] suggest to add a codec-switching impairment factor to the E-model as it improves the correlation of the E-model prediction to auditory test data with dynamically changing VoIP qualities to $r = 0.937$. In addition, Duran et al. [71] propose extensions for estimating the effect of time-varying speech quality. However, they do not validate its efficacy with subjective tests.

Regarding PESQ, Shiran and Shallom [180] also observe that a careful time-alignment is needed. Recently they propose to improve the time-alignment using Dynamic Time-Warping (DTW) instead of the standard correlation and splitting methods used in PESQ, which is one piece of the puzzle to account for the NGMN effects.

Goudarzi et al. [79] tested PESQ quality prediction capabilities in 3G networks. They observe that PESQ is a reliable tool for quality estimation in 3G in the context of narrowband PESQ only. Moreover, they do not account for mobility between heterogeneous networks and wideband-narrowband (WB-NB) codec changeovers.

6.5 Summary

In this chapter, we examine the quality prediction accuracy of WB-PESQ, a standardized signal-based quality prediction model, for NGMN scenarios. While packet loss is the most dominant factor for both WB-PESQ predictions as well as auditory MOS, codec changeover has significant effects as well. With regards to the predictive accuracy we illustrate that WB-PESQ underestimates the speech quality for some degradations which are typical for NGMN scenarios. These include codec changeovers between wideband and narrowband, signal fading during codec switching, and talk-spurt time-shifts due to jitter buffer size changes or underflow.

In summary, we highlight some inaccuracies of WB-PESQ predictions for NGMN-typical distortions. As a consequence, we believe that WB-PESQ needs to be improved to cope with the transmission variations in the modeling process.

7
Video QoE in NGMN

7.1 Overview

Due to the proliferation of smart mobile devices, a rapid growth in the consumption of video content is seen in the fixed and wireless areas [62]. In this chapter, we focus on video quality related to UDP based video transmission in Next Generation Mobile Networks. Recall, NGMNs interconnect heterogeneous wireless networks to an universal all-IP access platform. Once attached to the NGMN network, users can enjoy video and other services independently from the attachment point to the Internet. Through the support of mobility protocols, such as Mobile IP or SIP [177], technology independent roaming across all-IP networks is possible. For the video case, consequently, an adjustment of video stream parameters may be necessary to adapt the quality for the newly occurring networking circumstances, which becomes an intrinsic feature of NGMN systems.

The technology heterogeneity is reflected in different transmission layers. From the networking perspective, a trade-off between network coverage and link stability can be recognized. Networks of large coverage at cost of link stability, e. g., HSDPA, but also of small coverage and high link quality, e. g., WiFi, are connected to the system. From the video encoding perspective, various video encoding techniques can be applied. Well established codecs, such as H.264 or MPEG-4 Part 2 (abbr. MPEG-4) [198], differ in the compression ratio and robustness against transmission disturbances.

As a result, the concept of NGMNs faces new challenges in the service quality maintenance to be “always best connected”. User perception must be included in the mobility management and service adaptation rules to meet user expectations. In order to provide maximum quality, mechanisms like network handovers, video codec changeovers, bit-rate switching should be scheduled. However, thorough knowledge of the video quality perception by mobile users in NGMNs is still missing. More research to merge user perception and transmission conditions in NGMNs is required.

In this chapter, we attempt to close this gap. Focusing on the video transmission in NGMNs, we study the impact of the wireless access technology, network handover, video encoding and codec changeover, and bit-rate switching on the quality perception. We present a subjective experiment conducted to assess video quality perception in NGMNs. 50 NGMN conditions have been analyzed to identify perceptual bottlenecks of the NGMN networks and provide guidelines that help to include user perception in the mobility management of future wireless communication systems.

The rest of the chapter is structured as follows: Section 7.2 presents the QoE-Lab experimental setup for NGMN developed for this study. The test conditions and the subjective experiment are described in Section 7.3. The obtained results are presented and analyzed in Section 7.4. Section 7.5 discusses the related work. We summarize our work in Section 7.6.

7.2 QoE-Lab experimental setup for NGMN

We have already described key components of QoE-Lab testbed frame work related to NGMN in section 6.2.1. In this section, we describe changes in the testbed related to video quality assessment. Figure 7.1 depicts the key components of QoE-Lab testbed framework related to NGMNs. It consists of a Mobile Node (MN), a Corresponding Node (CN), a Home Agent (HA), and a Dummynet [170] traffic shaper. The MN includes the physical interfaces for two wireless networks WiFi, HSDPA, one wired DSL network, and a virtual interface created by the Mobile IP protocol. The CN is connected via a Gigabit Ethernet interface to the HA network. The CN and MN communicate via HA, and selectively via the Internet or the QoE-Lab. The HA is dual-homed, i.e. connected to both the Internet and the QoE-Lab; this measure was necessary, because the 3G UMTS/HSDPA access is available through commercial operators only. For controlled transmission impairments, 1) the *Dummynet* traffic shaper is interposed between the HA and access networks, and 2) the *netem* filter is applied on the HSDPA interface of the MN.

The CN and MN were deployed on Dell Optiplex 780 desktop computers: MN on Debian Linux 2.6.32, CN on Ubuntu Linux 2.6.28-19. The desktop computers were used to avoid any performance problems during simultaneous video processing and recording, network transmission and packet trace collection. The network traces were collected on all physical interfaces of MN and CN, and on the Mobile IP interface of the MN. For the mobility support ("make-before-break" policy), *lmip* client (a closed-source Mobile IPv4 implementation) has been installed on the MN. The HA was configured on a Cisco 7204 VXR router with Cisco IOS 12.1.

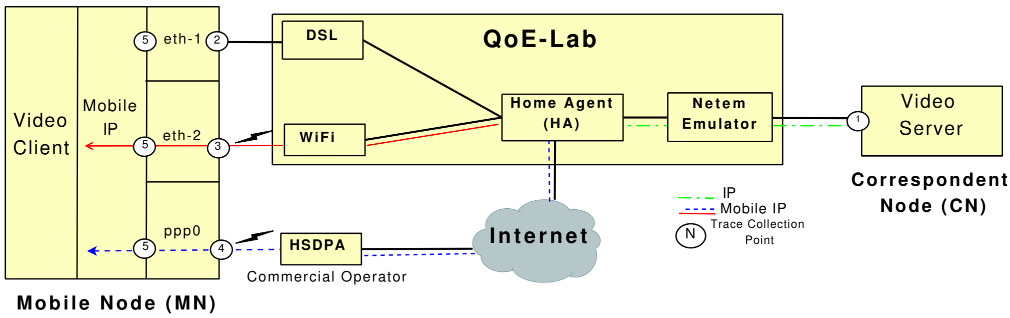


Figure 7.1: QoE-Lab experimental setup for the video services in NGMNs.

For the requirements of the research project, a video client has been developed based on the Voice over IP client (VoIP) framework PJPROJECT 0.8.3 [18]. The system architecture relies on SIP and RTP protocols [167]. The video data was transported by the RTP protocol, and packetized according to the codec-specific standards proposed in the respective RFCs. For video encoding, the *libavcodec* library (version from 06/07/2008) has been integrated into the client.

The video client allows for customizing of the transmission parameters. Four of them have been specifically set for the experiment. 1) The de-jitter buffer is an adaptive implementation, and was configured to the boundaries of 80 ms – 200 ms. 2) The H.264 packetization has been set to the non-interleaved mode. 3) The group of picture structure is configured to IPPPP [88]. 4) The Maximum Transfer Unit (MTU) size has been reduced to 1400 Bytes due to the overhead of the Mobile IP tunnel, and unnecessary packet fragmentation in the network.

In order to provide on-the-fly video codec changeover and bit-rate switching, a video codec changeover algorithm has been developed. Note that the H.264 SVC implementation is not used for this study. The codec changeover implementation relates to the soft voice codec switching presented in [194]. It renegotiates the codec during active transmission and synchronizes the display of the video frames received from two decoders during a changeover in the application. In this way, it was possible to reduce the visible artifacts of the codec changeover process.

7.3 Experiments

The source video material used for the generation of test stimuli was provided in the VGA resolution (640×480 pixels) with 25 frames per second (fps) and was stored in the uncompressed planar YUV 4:2:0 format. Four different videos sequences with three different content types have been used. Video 1 presents a 10 seconds extract of a movie. Video 2 contains 10 seconds of a soccer game sequence. Video 3 and Video 4 (9.5 s both) include the “head and shoulders” content of a men and a woman participating in a video call. The choice of the videos was made according to the typical content delivered to mobile devices and its impact on the video quality. The source material was fed to the CN, was encoded on-the-fly, transmitted to the MN, decoded, and recorded in the uncompressed format. This processing allowed to capture typical NGMN degradations coming from the network, but also their impact on the video application. Effects such as data loss in the jitter buffer or frame freezing could be captured.

The test sequences were either recorded without any alternation during the entire transmission, or included a mobility events, such as network, codec, bit-rate, or packet loss changes in the middle of the sample. Table 7.1 provides an overview of the transmission parameters under study. Three networks were included in the evaluation. While two wireless networks, HSDPA and WiFi, were representing the trade-off between network coverage and connection quality, the DSL network was applied as the reference transmission, as the quality is expected to be best. For the video encoding, two video codecs have been used, as they are well established for mobile communications and result in different encoding quality, and loss robustness. Three bit-rates 256, 768, 1536 kbit/s, were selected to cover low, medium, and high quality. Transmission impairments due to packet loss with loss values up to 5 % were applied for selected samples to emulate packet loss, as at this value the quality was already rated as "bad" for the VGA resolution and 25 fps during an internal pretest.

A viewing-only subjective experiment has been conducted to study the user perception. The test procedure was compliant with ITU-T Rec. P.910 [102]. According to the recommendation, the test sequences were presented sequentially to the test participants, and were alternated with the Absolute Category Rating (ACR) periods. The video quality was judged using an 11-point continuous rating scale according to [102]. 50 conditions formed a balanced set of test cases to cover the entire quality scale range. As a result, 200 stimuli were judged during one subjective experiment. The order of the stimuli was randomized for each of the test participants. 21 test subjects took part in the test, were aged between 18 and 40 years, and were balanced in gender. They were non-experts as they were not concerned with multimedia quality as part of their work. The resulting test duration of a single test was approx. 1 hour, with a 10 min. break in the middle of the test. Prior to the experiment, the test participants followed a training phase during which they judged a set of 15 stimuli, approximately covering the range of conditions of the later test. The test sequences were displayed on a Dell Inspiron 1012 netbook with a 10.1 inch WSVGA (1024×600 pixels) display, and the test participants were sitting in front of the notebook screen, with a viewing distance of approx. 50 cm.

Table 7.1: Transmission parameters of the video study.

Parameter	Values
Network	DSL, WiFi, HSDPA
Codec	H.264, MPEG-4
Bit-rate	256, 768, 1536 kbit/s
Loss rate (random)	0 ,1, 3, 5 %

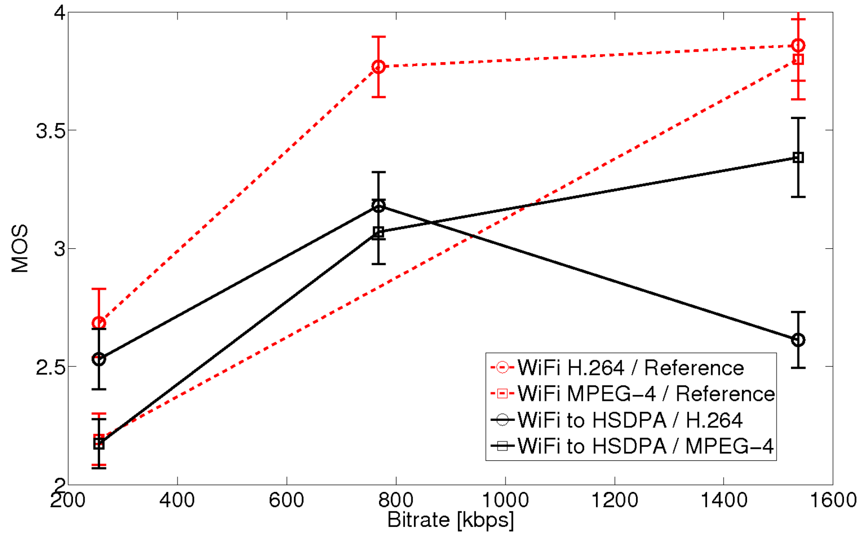


Figure 7.2: Network handover

7.4 Experimental results

The quality judgments obtained during the experiment described in the previous section have been linearly mapped to the 5-point ACR category scale [99]. The results are evaluated groupwise: network handover (NH), codec changeover (CC), bit-rate switching (BS), and loss compensation (LC). In the following subsections, the general behavior of relevant stable reference conditions is described first. The results are then compared to the NH, CC, BS, and LC conditions.

7.4.1 Network handover

According to the obtained quality scores, the access technology influences the quality perception. The video quality judged for reference H.264 at 1536 kbit/s in WiFi (3.85 MOS) is higher than in HSDPA (2.71 MOS). The same trend can be observed for H.264 at 768 kbit/s. The quality was rated with 3.76 MOS in WiFi and 2.99 MOS in HSDPA. The Analysis of Variance (ANOVA) with repeated measures and significance level of 5% confirmed the significance of these results with $F = 150, p < 0.001$. The MPEG-4 quality in WiFi and HSDPA (resp. 3.79, and 3.61 MOS) at the highest bit-rate of this study was not significantly ($F = 1.37, p = 0.079$) different.

Figure 7.3 compares the CC quality judgments with the reference conditions exemplary for the WiFi network. One can see that the switching from H.264 to MPEG-4 always degrades the quality. The impact of CC is inverse proportional to the encoding bit-rate. The lower the bit-rate the higher the quality deterioration. Switching in the opposite direction, i.e. from MPEG-4 to H.264, improves the quality perception for 256 kbit/s and degrades the quality

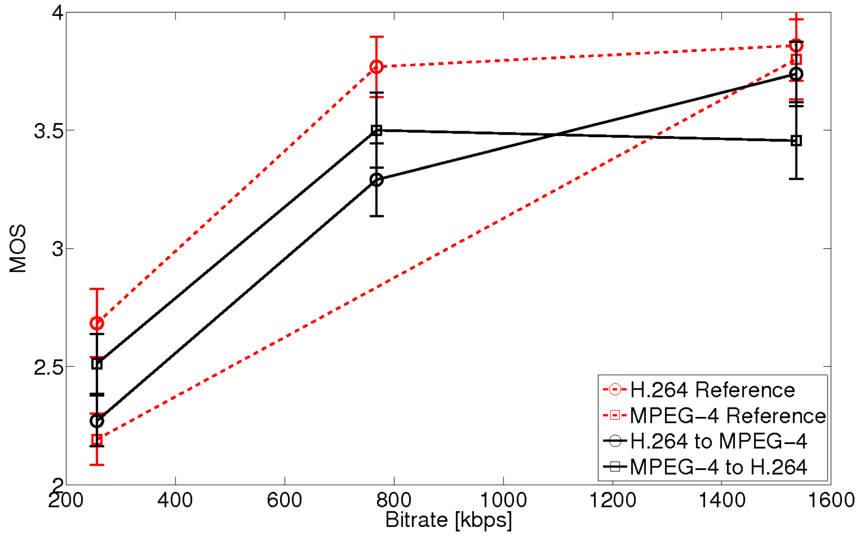


Figure 7.3: Codec changeover in WiFi

for 1536 kbit/s (potentially because of the CC operation and similar quality provided by both codecs at 1536 kbit/s).

7.4.2 Bit-rate switching

A significant ($F = 50.3, p < 0.001$) perceptual influence of the applied H.264 bit-rates could be found. However, a Post-hoc analysis (Scheffé) confirmed that the quality provided by reference H.264 in WiFi at 768 and 1536 kbit/s do not significantly differ from each other. The largest quality difference perceived for H.264 in WiFi was between the 256 and 1536 kbit/s (resp. 2.68 and 3.85 MOS). A similar effect of the bit-rate and video quality could be observed for MPEG-4 in WiFi. The quality at 1536 kbit/s (3.79 MOS) was higher than for 256 kbit/s (2.19 MOS). This difference was more visible than for the H.264 encoder. In the HSDPA network, the MPEG-4 bit-rates of 768 and of 1536 kbit/s were rated in a proportional way (resp. 3.23 and 3.61 MOS). An opposite trend in HSDPA was observed for H.264. The quality score for 768 kbit/s (2.99 MOS) was higher than (2.71 MOS) obtained for 1536 kbit/s; network analysis will follow for explanation.

Figure 7.4 compares the quality scores obtained for the reference conditions and for bit-rate switching in the WiFi network. The BS generally increases or decreases the quality in accordance with the switching direction. An exception could be observed for the H.264 switching from 768 kbit/s to 1536 kbit/s, which relates to the aforementioned quality similarity for H.264 at 768 and 1536 kbit/s. This quality drop is potentially caused by the BS operation. Switching the H.264 bit-rate from the highest to the lowest degrades the quality perception and is similar rated as the reference case H.264 at 256 kbit/s. A quality improvement of 0.45 MOS is possible by switching H.264 from 256 kbit/s to 1536 kbit/s.

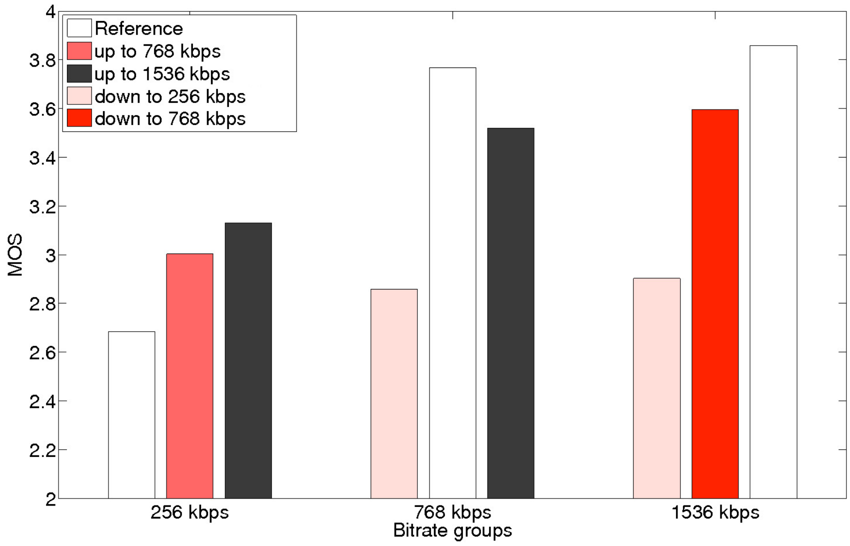


Figure 7.4: Bit-rate switching for H.264 in WiFi

However, the perceptual gain is less relevant than the aforementioned degradation in the opposite direction.

7.4.3 Loss compensation

NH, CC, or BS might be advantageous for reducing deteriorating effects caused by packet loss in the wireless network. This is analyzed in this section for the WiFi network and the H.264 codec at a starting bit-rate of 1536 kbit/s. Figure 7.5 presents the quality judgments obtained for 0%, 1%, 3%, and 5% of random packet loss with and without any loss adaptation measures. Packet loss degrades the user perception in a significant way ($F = 231,\ p < 0.001$). Already the packet loss of 1% causes a quality drop of 1.71 MOS and, as presented in [93], exhibits an exponential quality degradation trend.

The usage of the aforementioned adaptation techniques does not provide any benefits for a loss-free transmission. In this case, all three methods decrease the quality perception (see preceding sections). However, in the case of packet loss, all the adaptation mechanisms improve the video quality. For the lowest packet loss value of 1% the adaptation techniques allowed to obtain a quality gain of 0.19 MOS, 0.17 MOS, and 0.39 MOS by NH, BS, CC, respectively. For the highest packet loss condition of 5%, a quality improvement of 0.34, 0.31, 0.47 MOS by NH, BS, CC could be observed, which is a significant result ($F = 7.97,\ p = 0.001$). A Post-hoc analysis (Scheffé) shows that the compensation by NH and BS do not significantly differ from each other ($p = 0.97$) and CC significantly outperforms the other measures (resp. $p = 0.007,\ p = 0.004$) and is the most effective technique of video quality adaptation in a lossy environment.

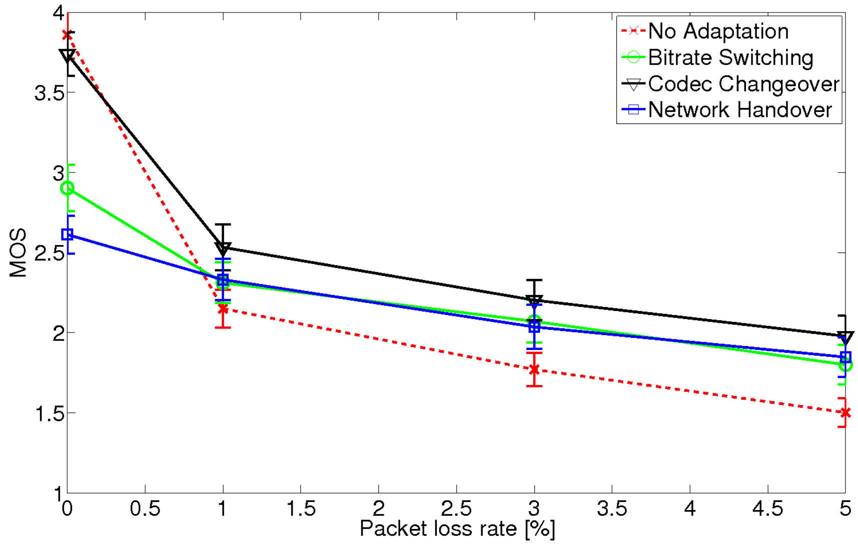


Figure 7.5: Loss compensation (starting in WiFi, H.264@1536 kbit/s)

7.5 Related work

Evaluating video Quality of Experience (QoE) is important due to its increased consumption in fixed and mobile networks. To assess the video quality, two methodologies, subjective and objective evaluation are available as explained in Section 2.5.2. For the subjective quality judgments [102], the quality of video material is judged by human viewers. Even though subjective testing is the most valid measure, it is an expensive and time-consuming way of quality assessment. To overcome this problem, objective quality metrics have been developed. These objective quality models capture the quality degradation in terms of technical parameters, such as Peak Signal to Nose Ratio (PSNR) and Structural Similarity (SSIM) index [200]. While PSNR and SSIM metrics provide video quality predictions in the presence of full reference (FR) or reduced reference (RR), quality estimations for live video streams, e. g., IPTV can be obtained directly from video bitstreams [188]. However, these models have not been validated for NGMN conditions.

The quality of video transmission for different wireless environments has already been addressed extensively in the literature. The video quality degradations due to distance, obstacles and motion have been quantified in [73]. Piamrat et al. [160] have presented a model for quality assessment in wireless networks. Throughput problems during video transmission due to network handover is studied by [164]. Adaptive media streaming in heterogeneous wireless networks is explored in [174].

The video quality can also be adapted in wireless networks by video bit-rate adaptation. As presented in [115], a Quality of Experience (QoE) driven model can help to adjust the sending bit-rate according to the video content. Additionally, the need of video bit-rate adaptation for UMTS networks is presented in [40]. The realization of scalable video encoding in heterogeneous networks is also shown in [205]. The authors developed a non-reference quality prediction model that is deployed at the user-end and controls the scalable video encoding (SVC) according to the blurriness, blockiness, and jerkiness of the video stream.

Even if the instruments of mobility management in NGMN transmission have been already addressed in the past, not all perspectives of user perception in NGMNs have been thoroughly explored yet. The impact of NGMN events on video quality perception still needs to be analyzed, as it already has been done for voice services in [142].

7.6 Summary

Next Generation Mobile Networks provide a heterogeneous all-IP platform for video service delivery with mobility support. This concept faces new challenges in the maintenance of high video quality and requires integration of the user perception in the mobility management.

In this work, we present an evaluation of the subjective video experiment with 50 transmission conditions in NGMNs. They are arranged in groups to assess the impact of 1) the access technology and network handover, 2) video codecs and codec changeover, 3) video bit-rate and bit-rate switching, and 4) to provide guidelines for packet loss adaptation. The main conclusions for the test set-up we used are as follows:

- The access technology has a significantly impact on the video quality provided by H.264, and a non-significant influence on the MPEG-4 transmission.
- Network handovers from WiFi to HSDPA always degrade the quality proportionally to the encoding bit-rate. The MPEG-4 codec is less sensitive for the handover impairment than H.264.
- In a loss-free environment switching the video codec generally degrades the quality inverse proportionally to the bit-rate. However, a quality improvement can be achieved by switching from MPEG-4 to H.264 at 256 kbit/s.
- Switching the video bit-rate improves or decreases the video quality in accordance with the switching direction. Downgrading the bit-rate has stronger impact on the user perception than upgrading.
- If packet loss occurs, a codec changeover is the best packet loss adaptation technique, and significantly outperforms the network handover and bit-rate switching.

8

Cross-Layer effects on video QoE in NGMN

8.1 Overview

The increasing traffic volume by real-time entertainment on the mobile devices as discussed in Section 2.1.1 attracted our attention towards evaluating user experience for the web video streaming and its interactions with TCP. Indeed, wireless Internet access has grown significantly over the last decade from GRPS and EDGE, to UMTS/HSDPA, UMTS/HSPA+, WiFi, WiMax and towards 4G LTE Advanced, promising ever more bandwidth to the users. Due to this multiplicity of choices, the NGMN alliance makes several recommendations including seamless mobility and roaming over different networks, end-to-end Quality of Service (QoS), and real-time and streaming support [143]. These recommendations follow user expectations, as today's users enjoy smart phones, which are equipped with multiple wireless network interfaces (e.g., UMTS or WiFi) and expect to be able to choose the right network according to their own personal preferences, network quality, and cost [148]. Additionally, Internet services, e.g., web video streaming, are growing in popularity among mobile users. One of the popular sites, YouTube Mobile, reports more than 100 million video playbacks per day [26].

Motivated by their popularity, in this chapter we focus on Quality of Experience (QoE) in video streaming applications. These applications use pseudo-streaming and the content is delivered using HTTP/TCP [173]. Our goal is to understand the impact of time-varying channel characteristics in heterogeneous wireless networks on web video streaming QoE.

The ideal way to assess video QoE is to perform perception tests. These tests help calculate a Mean Opinion Score (MOS), which expresses the mean quality score of a group of users according to ITU-T Rec. P.800 [99]. However, such an evaluation requires time-consuming and expensive subjective tests. Due to these limitations, we follow the objective video quality assessment approach and analyze the expected user experience by both a QoE metric – PSNR (Peak Signal to Noise Ratio) and QoS metrics such as throughput, and delay under varying network conditions. Furthermore, to expose the cross-layer dynamics, we use TCP statistics which also enables us to correlate QoS and QoE behavior.

Our measurements include two types of access technologies: UMTS/HSDPA and WiFi, which show different link quality behavior over time. The main contribution of our work is an in-depth cross-layer QoS and QoE analysis of web video streaming across these technologies. Our results show that:

- **Better video QoE can be achieved with WiFi.** WiFi communication is limited by interference but shows more stable behavior over time. However, the transmission quality depends significantly on the concurrent demand on the wireless channel.
- **Video QoE in HSDPA is more sensitive to network dynamics.** The sensitivity stems from different reasons including transport layer interactions, scheduling and ARQ (Automatic Repeat Request) mechanisms.
- **The congestion control mechanisms used by TCP have a huge impact on the QoE.** We use *CUBIC* TCP [82], which is the default TCP variant used in todays popular mobile devices, e.g., Android phones. However, CUBIC TCP is designed for high speed networks and employs more aggressive congestion control, which in turn results in less stable communication, especially for HSDPA.
- **Handover from WiFi to HSDPA degrades performance.** Experiments show that WiFi to HSDPA handover degrades video QoE, while HSDPA to WiFi handover immediately improves QoE.

The rest of the chapter is organized as follows. In Section 8.2, we describe our NGMN testbed. We discuss our experimental methodology in Section 8.3. In Section 8.4, we present the performance results. Section 8.5 presents related work and we summarize and discuss future work in Section 8.6.

8.2 QoE-Lab experimental setup for video streaming

To understand the factors that affect QoS and QoE of web video streaming in NGMNs, we again rely on our QoE-Lab testbed framework. The main components of our integrated testbed is shown in Figure 8.1, which consists of the Mobile Node (MN), the Correspondent Node (CN) and the Mobile IPv4 Home Agent (HA). The MN acts as the video client and the CN as the video server. All communication between the CN and the MN is managed by the HA. A *netem* network emulator exists between the CN and the HA to emulate different packet loss rates.

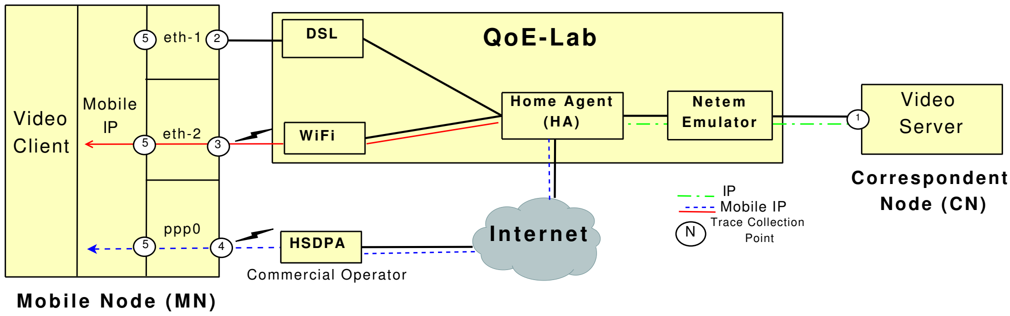


Figure 8.1: QoE-Lab experimental setup for the video services in the NGMN.

The CN and MN are laptops running Ubuntu Linux 2.6.28.16. The HA was configured on a *Cisco 7204 VXR* router with *IOS 12.1*. The MN has a WiFi, a HSDPA, and a 1 Gbps Ethernet interface. An additional virtual interface is created by the Mobile IP protocol. The CN is connected to the HA through 1 Gbps Ethernet, passing through the *netem*. The HA is dual-homed, i.e., connected to both the QoE-Lab and the Internet. We connect the HA to the Internet since HSDPA access is available through commercial operators only. The WiFi access point was configured as a standard IEEE 802.11g router with the transmission rate of 54 Mbps. The HSDPA connection as provided by a large European service provider operates at 7.2 Mbps for downlink.

For the mobility support, we rely on a "make-before-break" policy using *lmip*, a closed-source implementation of a Mobile IPv4 client. We use *lmip* at the MN, which allows the MN to perform network handovers during on-going sessions. Note that, while there is IPv4/TCP communication between the CN and the HA, the communication between HA and MN goes through a Mobile IPv4 UDP tunnel (port 434) irrespective of the selected access network. This constitutes a typical hybrid transport layer communication scenario while roaming.

We collect experiment traces by using *tcpdump* on all the physical and virtual interfaces of CN and MN. To monitor the dynamics of TCP, we use *tcp-hook* [204], a Linux kernel module based on the In-kernel Protocol Sniffer (IPS). Mainly, it provides a hook between TCP and the network layer. We deploy *tcp-hook* on the CN to observe TCP congestion control dynamics such as TCP congestion window size and estimated round trip times at the video server. We repeat the experiments multiple times with different packet loss rates.

8.3 Video quality evaluation in NGMN

In this section we present our experiment methodology for our video streaming evaluation in NGMNs. Next, we discuss the metrics used for evaluating QoS and estimating QoE for web streaming.

8.3.1 Experiment methodology

To create a web streaming environment, we install the TCP-based Tribler video streaming system [23] in the QoE-Lab testbed. A 10 minute movie-sequence [7], encoded in H.264 at 24fps (VGA-resolution), is streamed from the CN to the MN. On the video server (CN), we use the CUBIC TCP congestion control algorithm, which is the default in Linux distributions as well as Android phones. The key feature of CUBIC is that the congestion window growth relies on the real time between two congestion events [82]. It uses a cubic function for its window adjustment algorithm, hence its name. In addition, we use the default Linux settings, which include TCP window auto-tuning to dynamically adjust TCP's sender and receiver windows to better use the link capacity.

To characterize the performance, we vary the following:

- **Access technology:** We associate the MN to WiFi and HSDPA networks in separate experiment instances. Note that WiFi is a simple technology, which uses CSMA/CA to grant access to the shared wireless medium. In contrast, HSDPA uses more complex mechanisms such as *fast link adaptation* to continuously adjust the modulation and coding scheme, *packet scheduling* that exploits the link quality information, and *Hybrid Automatic Repeat Request (HARQ)* and *soft combining* to let terminals recover from errors by explicitly requesting retransmissions and exploiting parts of erroneous frames [65]. These differences are expected to have an impact on transmissions and hence, web video streaming QoE.

- **Packet loss rate:** As packet loss tends to be the dominant reason that affects multimedia quality, we inject additional 0-15% random packet loss in the video streams using *netem*.

- **Existence of network handovers:** To test vertical network handovers and the effect of "make-before-break" policy on QoE, we force the MN to associate from WiFi to HSDPA and vice versa during an ongoing video transmission.

- **Video pre-roll delay**: The pre-roll delay is the buffering time at the client's video decoder before playing back the video. Higher values delay the video start time but improve user perception. The pre-roll delay is varied from 0 to 20 s.

8.3.2 Video QoS and QoE

To capture the user expectations in mobile multimedia delivery, it is necessary to understand both QoS and QoE guarantees of NGMN [34]. To estimate the QoE of the video stream,

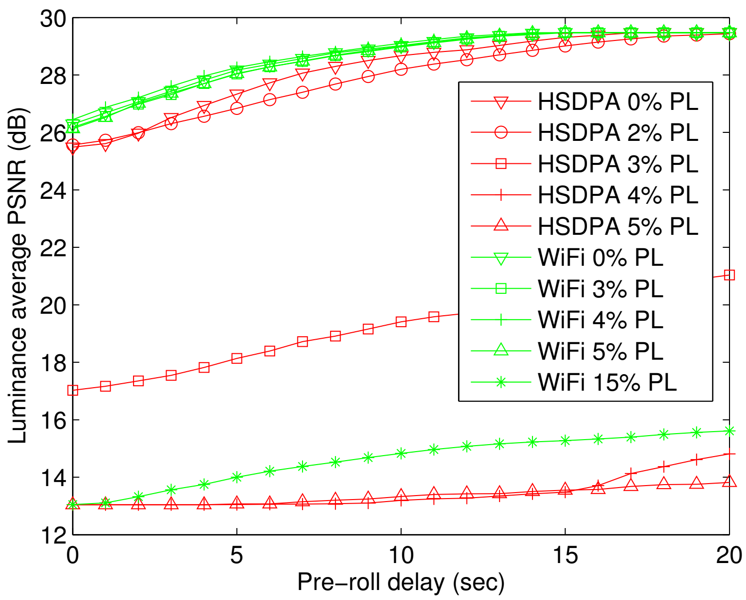


Figure 8.2: PSNR (dB) vs. pre-roll delay (sec) at MN in WiFi and HSDPA

we use the most commonly used metric in objective quality evaluations to estimate the user perception: PSNR (Peak Signal to Noise Ratio). We measure the PSNR and calculate it for different pre-roll delays between the original and the received video clip at the video client on the MN [33].

The video decoder uses copy-previous error concealment to compensate lost parts of the stream. If the video stream has N frames, the Mean Squared Error (MSE) for frame F_n $(0 \leq n < N)$, given that F'_n is displayed instead of F_n by the decoder, is

$$M_n = \frac{1}{X \cdot Y} \sum_{x=1}^{X} \sum_{y=1}^{Y} [F_n(x,y) - F'_n(x,y)]^2, \tag{8.1}$$

where the frame size is $X \times Y$ pixels. Denoting M as the average M_n for N video frames,

$$M = \frac{1}{N} \sum_{n=0}^{N-1} M_n. \tag{8.2}$$

the average quality in terms of PSNR, in dB, is then computed as:

$$PSNR = 10 \cdot log_{10} \frac{(255)^2}{M}, \tag{8.3}$$

where 255 is the maximum luminance value of a pixel for 8-bit pictures.

To complement the QoE metric and to understand the corresponding network level QoS, we also compute throughput, delay, and TCP statistics such as RTO (Retransmission Time-Out), RTT (Round Trip Time) to ACK TCP segments, number of lost segments, duplicate

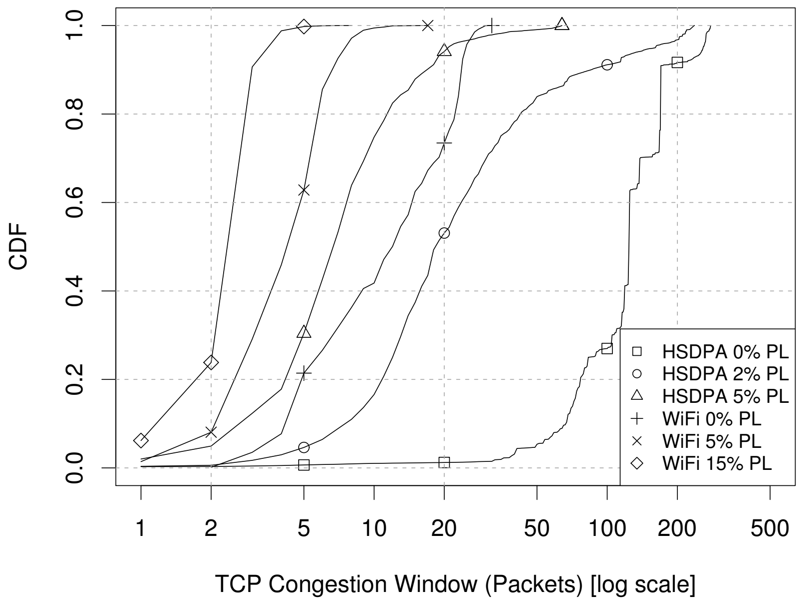


Figure 8.3: TCP congestion window at CN in WiFi and HSDPA.

acknowledgements, and fast retransmissions. Analyzing information from different layers provides insight into the QoS for different access technologies and NGMN conditions. Next, we present our performance evaluation in terms of these QoS and QoE metrics.

8.4 Results

In this section we present our evaluation of quality of web video streaming in NGMN. We first discuss the QoS and QoE performance in WiFi and HSDPA networks separately. We also take a deeper look at TCP performance in WiFi and HSDPA, which illustrate the importance of understanding cross-layer trade-offs when evaluating video QoE in NGMNs. Finally, we conclude with a discussion of impact of network handovers.

8.4.1 Web video streaming performance with WiFi and HSDPA

We first evaluate video QoE in terms of PSNR in WiFi and HSDPA networks with different packet loss rates. We also vary the pre-roll delay to understand its effect on improving QoE. We observe that the performance is significantly different for the *same* packet loss rate depending on the access technology (WiFi or HSDPA). Fig. 8.2 shows that WiFi generally achieves higher PSNR compared to HSDPA. For instance, with 5% packet loss, the video quality remains high for WiFi (above 26 dB). However, with HSDPA and the same loss rate, the PSNR values become unacceptable. Such low quality is experienced for WiFi only with

Table 8.1: Average throughput and TCP statistics for WiFi and HSDPA for varying packet loss.

Network	Loss (%)	Average Throughput and Standard deviation (kbps)	Lost Segments	Duplicate ACKs.	Fast Retransmits	Avg. # of RTOs per 1 s	% of intervals with RTO
WiFi	0	654 ± 230	17	131	11	1	0.917
	2	632 ± 243	663	1745	314	2.10	25
	5	661 ± 228	1752	4726	969	2.13	55.12
	10	680 ± 307	2793	8668	1932	2.01	65.64
HSDPA	0	698 ± 415	234	310	3	3.66	9.67
	1	655 ± 296	393	4628	78	2.73	17
	2	330 ± 155	518	3757	113	3.23	30.92

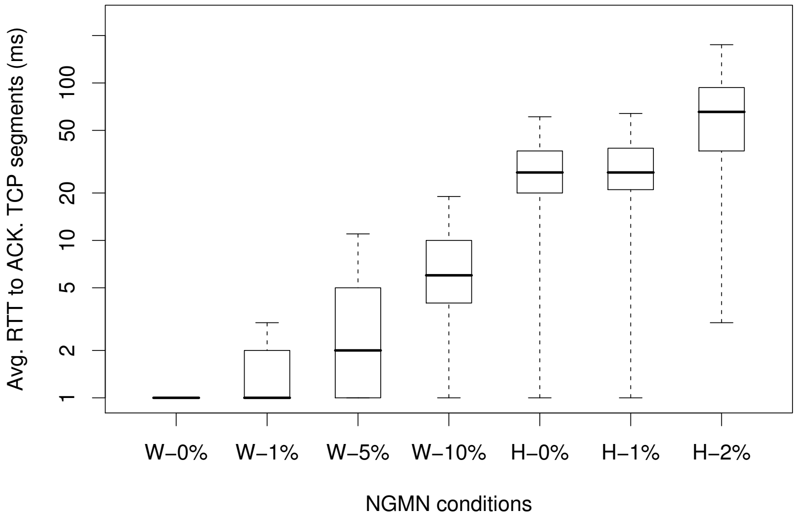


Figure 8.4: Average RTT to ACK TCP segment variations in 1s bin, W:WiFi, H:HSDPA

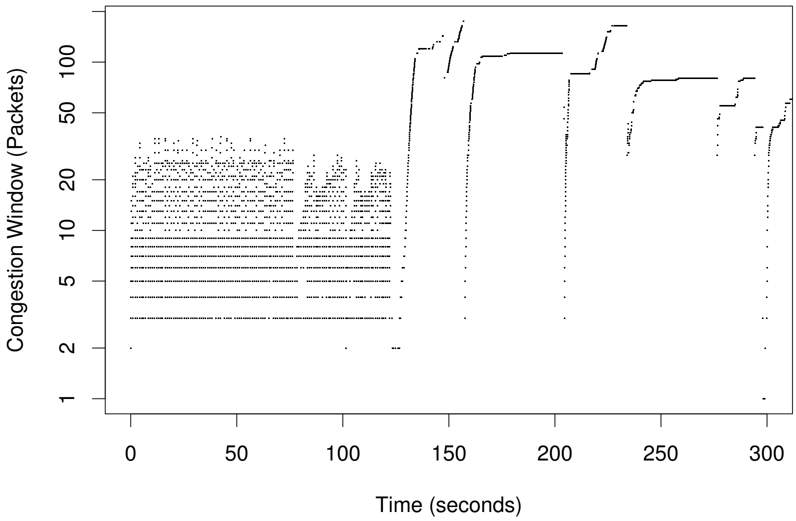


Figure 8.5: Congestion window vs. time, WiFi to HSDPA handover at 125 s, Loss=0%

loss rates higher than 15%. Also, for a fixed PSNR value, WiFi requires shorter pre-roll delay (e.g., at 28 dB, pre-roll delay of $\approx 5s$ for WiFi and $\approx 10s$ for HSDPA) and therefore

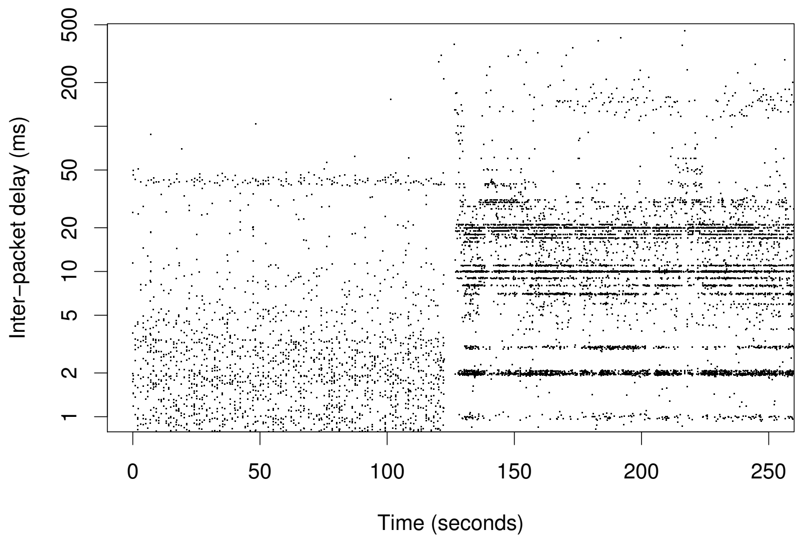


Figure 8.6: Inter-packet delay vs. time, WiFi to HSDPA handover at 125 s, Loss=0%

provides a better QoE.

Table 8.1 illustrates the same behavior from the network QoS point of view. While WiFi maintains a high throughput even with 10% loss rate, the throughput of HSDPA degrades immediately with 2% packet loss. Furthermore, for 0% loss rate, HSDPA exhibits less stability compared to WiFi with significantly higher standard deviation. This is also captured by the PSNR results, which point at unacceptable web video streaming qualities with HSDPA. We identify that the poor performance of HSDPA is due to the time varying channel availability due to shared downlink and higher RTTs that undermine TCP performance. We next explain these cross-layer effects in more detail.

8.4.2 A closer look at TCP and lower layer interactions

Next, we discuss TCP statistics that clarify the performance differences between WiFi and HSDPA. In CUBIC TCP, when RTTs are short, the window growth rate could be lower as compared to traditional TCP due to the fixed window growth rate [82]. We observe this effect in WiFi, which experiences shorter RTTs and hence maintains smaller congestion windows compared to HSDPA. Fig. 8.4 shows the variations in average RTT to ACK TCP segments in 1 s bin. For WiFi RTTs are much shorter than those for HSDPA across all packet loss rates. Furthermore, the HSDPA RTT values show a higher variance at all packet loss rates hinting at the use of more complex link layer mechanisms (e.g., HARQ). Based on these RTT values, we calculate the bandwidth-delay product (BDP) for 0% packet loss, and get ≈ 7 KB for WiFi and ≈ 45 KB for HSDPA. This difference in BDP is also evident from the congestion window evolution (see Fig. 8.3). While WiFi congestion window sizes lie between $5-20$, HSDPA shoots its congestion window up to ≈ 250. However, as shown earlier, this does not necessarily lead to high throughput due to aggressive congestion window evolution.

Table 8.1, where total number of losses, duplicate ACKs and fast retransmits are also reported, sheds more light on the difference in congestion window evolution in WiFi and HSDPA. In WiFi packet losses are higher. However, the majority of the losses are recovered during fast retransmit even for high loss rates (e.g., 10%). Still, these packet losses result in reducing the window size by a factor of $\beta = 0.2$ [82] and hence limit congestion window growth. In contrast, in HSDPA, the number of lost segments are lower (e.g., for 2% packet loss, the number of lost segments is 518 for HSDPA, and 663 for WiFi). However, the number of fast retransmissions is also significantly lower (e.g., 113 compared to 314, in HSDPA and WiFi). We infer that since the number of duplicate ACKs is high but the number of fast restransmits is low, the number of 3 consecutive duplicate ACKs is low in HSDPA. Hence, the high number of duplicate ACKs for HSDPA is mainly due to packet re-ordering and not packet loss. Essentially, as a result of the re-transmission activity of HARQ processes in HSDPA, subframes may arrive out-of-sequence. This is evident from the TCP traces, however, not possible to verify directly due to the use of commercial UMTS service.

Furthermore, we analyze the average number of retransmission timeouts (RTO) and how often RTOs occurs. Here, the total experiment time is divided into 1 s intervals, and the average number of retransmission timeouts is calculated for the intervals when at least one timeout occurs. TCP over HSDPA experiences more timeouts at a given interval and also, the percentage of intervals with timeout events increase as the loss rates increase. On the other hand, especially for the 0% loss case, WiFi experiences significantly less timeouts. Due to these combined effects, WiFi is able to maintain better video QoE.

8.4.3 The effect of vertical network handovers

Finally, we show an example case where we force a handover from WiFi to HSDPA at 125 s. As we discussed in Section 8.2, we use a “make-before-break” policy. Figs. 8.5 and 8.6, depict the congestion window evolution and inter-packet delay with time, respectively. The results indicate that “make-before-break” policy interrupts web video streaming minimally and the user perception is not expected to be affected. Furthermore, in the graphs, the trends associated with each radio access technology is clearly visible indicating that the user perception is mainly determined by the associated access technology. Our results show that users should take advantage of WiFi networks to get better QoE for web video streaming applications, which are sensitive to high variations in throughput and retransmission timeouts. For HSDPA, CUBIC TCP which is a TCP variant for high speed networks, is not a good choice and degrades performance.

8.5 Related work

The performance comparison of 3G and WiFi networks has gained recent interest in the research community. For instance, in [66, 76], the potential of opportunistic use of WiFi networks to reduce the load in 3G networks is investigated. Both works show that, in mobile scenarios, WiFi experiences frequent disconnections but it can provide higher throughput. On the other hand, 3G has stable coverage but offers lower throughput. Additionally, several works evaluated QoS/QoE for multimedia applications in either 3G or WiFi networks. In [112], HSDPA user experience was studied for HTTP/TCP and VoIP applications in a live network. It was shown that maximum capacity of the link is reached with relatively larger payloads and TCP performance is dependant on both uplink and downlink performance. In [61], the impact of MAC layer local retransmission mechanisms in 3G networks are listed as increased delay and rate variability for TCP. Similar to our results, performance degradation due to the aggressive congestion window evolution of CUBIC TCP was noted for WiMAX networks in [84].

Improving multimedia quality in NGMNs is also an active area of research. In [31], a cross-layer optimizer was proposed, which uses information about MAC layer conditions to perform video streaming rate adaptations (i.e., codec switching). Codec switching is also supported in [19, 151]. IEEE 802.21 [10] work group focuses on primarily network (vertical) handovers and enables co-operative handover decision making between clients and networks. Several works [57, 71, 80] study performance impact of vertical handovers for multimedia traffic in heterogeneous networks, typically via simulations.

In this chapter, we complement these related works by using both QoS and QoE metrics as well as TCP statistics to understand the web video streaming performance for different access technologies in real life conditions. These results provide a deeper understanding for the cross-layer affects between MAC and transport layers, and the conditions that bring gains from network switching.

8.6 Summary

Adoption of mobile web streaming will be dependent on the quality that the users receive as they roam between different networks. In this chapter, we present a QoE study of web video streaming in NGMN scenarios. Our results can effectively be used for policy and decision making strategies such as video codec change-over or bit-rate adaptation to improve the video quality. Essentially, we show that with two different access technologies, WiFi and HSDPA, the QoS and the QoE performance is strictly tied to the interactions between the underlying MAC layer and the transport layer mechanisms. The default kernel settings (i.e., TCP Cubic variant with auto-tuning) are not recommended for HSDPA networks and call for cross-layer adaptation mechanisms, where TCP congestion control as well as the local ARQ mechanisms such as HARQ at the link layer are selectively used depending on the web video streaming QoS and QoE.

9 Conclusion and outlook

In this thesis, we advocate that technologies, devices, and changing trends in the Internet landscape can cause novel network effects and can have an adverse impact on network performance and end-user perception. We thus need to understand both the qualitative and quantitative impact of new network effects on the various applications from end-user point of view. We argue that different research areas namely networking and quality engineering are indispensable to tackle this task. In this chapter, we summarize our work, discuss key lessons learned from our work and outline future work.

9.1 Summary

QoE-Lab testbed framework

We begin by presenting the design and architecture of QoE-Lab, a modular testbed for the evaluation of the network performance and Quality of Experience (QoE) of different applications under the heterogeneous networking conditions. It features controlled background traffic generation, network emulation, seamless mobility between different access technologies, high precision monitoring, and an interconnection between experimental setup and the real Internet. It also includes hardware that is used in operational settings as well as emerging virtualization technologies, including OpenFlow. We describe our software tool, EXPAUTO, which is used for creating different test scenarios, allocating resources, monitoring and collecting data at various networking layers.

Leveraging the strength of QoE-Lab framework, we present two QoE case studies. The first case study explores the impact of different load intensities on IPTV video quality. We find that video QoE is much more sensitive to network traffic load when compared to audio QoE.

The main reason is that bursty packet losses within the video stream affect multiple frames resulting in poor visual quality, whereas in audio, losses are more easily concealed and recovered. Our second case study examines the impact of background traffic on emerging virtualization technologies and OpenFlow architectures. We find that offline migration of video servers has more impact on video quality as compared to live migration. In the case of an OpenFlow controlled network, quality is not a direct function of the network load rather it depends on the flow arrival rate. Furthermore, purely reactive controller with a fine-grained flow model can be highly problematic for QoE when faced with realistic, bursty background traffic.

Packet loss process under different load intensities and buffer sizes

Through controlled experiments in QoE-Lab, we study the relationship between several parameters, including load, router buffer size, and flow size distribution, on the properties of traffic and the loss process. Our sensitivity analysis reveals that small buffers have a deep impact on the ability of TCP to use the link capacity. We confirm that both high load and small buffers lead to high packet losses. However, small buffers have a much higher impact on losses than high load. Surprisingly, we find that packet losses do not affect all flows similarly. Irrespective of the network load and the buffer size, there are few unhappy flows, especially small ones, that observe unusually large losses. On the other hand, most flows, especially large ones, are happy and do not observe high losses compared to the global loss rate. Furthermore, very few flows actually observe a loss rate similar to the average loss rate. Therefore, any single flow is very unlikely to observe the global packet loss process. Finally, our study of the packet loss process reveal that it can exhibit scaling properties under high load as well as significant irregularities under large buffer sizes. When the buffer size is very small, the loss process is uncorrelated at time-scales below the typical RTT.

Flow performance in the Internet

The observations from our controlled experiments regarding packet losses lead us to examine flow performance in the Internet. To this end, we study flow-level performance of popular applications across flow sizes. Our metrics to gauge flow performance include retransmissions, out-of-sequence packets, throughput, and round-trip times. Based on traces gathered from three very different network environments, we compare flow performance through the lens of these metrics, under different network loads, access link capacities, and traffic directions. We find that network load has a significant impact on flow performance, and that different applications are impacted differently. We also find that flow performance varies significantly across flow sizes. For example, contrary to popular belief, small flows, that make up a majority of the flows, experience significant retransmission rates, across all applications. Large flows on the other hand, although fewer in number, can experience limited retransmissions. We observe a marked contrast between HTTP and P2P flows. Indeed, P2P flows suffer from continuously high retransmissions compared to HTTP. We identify the access part of the network as the area of the network responsible for these retransmissions, not the network core. Moreover, part of the bad performance of P2P is related to spurious retransmissions.

VoIP QoE prediction in NGMN

Our flow-level analysis encompasses performance results of a large user population of the Internet. We next turn our focus towards QoE studies in NGMN with real users. For this purpose, we first examine the quality prediction accuracy of WB-PESQ, a standardized signal-based quality prediction model, for NGMN scenarios. Mobile manufacturers often rely on PESQ model to gauge the call quality of new generations of phones. Therefore, any inaccuracies in the PESQ model may result in substandard selection of new generation of phones. While packet loss is the most dominant factor for both WB-PESQ predictions as well as auditory MOS, codec changeover has significant effects as well. With regards to the predictive accuracy we illustrate that WB-PESQ underestimates the speech quality for some degradations which are typical for NGMN scenarios. These include codec changeovers between wideband and narrowband, signal fading during codec switching, and talk-spurt time-shifts due to jitter buffer size changes or underflow. We highlight some inaccuracies of WB-PESQ predictions for NGMN-typical distortions. As a consequence, we believe that WB-PESQ needs to be improved to cope with the transmission variations in the modeling process.

Video QoE in NGMN

Results of our VoIP case study in NGMN motivated us to further look for video QoE in NGMN. To this aim, we perform subjective video quality evaluation in NGMN. We select 50 transmission conditions and arrange them according to the groups to assess the impact of 1) the access technology and network handover, 2) video codecs and codec changeover, 3) video bit-rate and bit-rate switching, and 4) to provide guidelines for packet loss adaptation. Our major findings from this study are: i) The access technology has a significantly impact on the video quality provided by H.264, and a non-significant influence on the MPEG-4 transmission, ii) Network handovers from WiFi to HSDPA always degrade the quality proportionally to the encoding bit-rate. The MPEG-4 codec is less sensitive for the handover impairment than H.264 , iii) In a loss-free environment switching the video codec generally degrades the quality inverse proportionally to the bit-rate. However, a quality improvement can be achieved by switching from MPEG-4 to H.264 at 256 kbit/s, iv) Switching the video bit-rate improves or decreases the video quality in accordance with the switching direction. Downgrading the bit-rate has stronger impact on the user perception than upgrading, and v) If packet loss occurs, a codec changeover is the best packet loss adaptation technique, and significantly outperforms the network handover and bit-rate switching. Overall, this study contributes to the integration of user perception in the mobility management in future mobile networks

Cross layer effects on video QoE in NGMN

Finally, we present a QoE study of web video streaming in NGMN scenarios. Our results can effectively be used for policy and decision making strategies such as video codec changeover or bit-rate adaptation to improve the video quality. Essentially, we show that with two different access technologies, WiFi and HSDPA, the QoS and the QoE performance is strictly tied to the interactions between the underlying MAC layer and the transport layer mechanisms. The default kernel settings (i.e., TCP Cubic variant with auto-tuning) are not recommended for HSDPA networks and call for cross-layer adaptation mechanisms, where TCP congestion control as well as the local ARQ mechanisms such as HARQ at the link layer are selectively used depending on the web video streaming QoS and QoE.

9.2 Discussion

Motivated from the observations of two decades of Internet traffic characterization studies that the Internet traffic is bursty in nature, we set out to find the impact of bursty traffic on multimedia applications. We define our goal to investigate how much burstiness in the traffic impact different applications and how much of this impact is perceived by the end user. The formulation of our research questions have two contrasting dimensions rooted deeply in the networking and quality engineering research communities.

To our surprise, we found that these two communities work with different assumptions. We believe that both communities should include each other's view point in their studies for better overall understanding of users experience about applications and its relation to the network. For instance, a great deal of networking research use quality assessment in merely objective manner by relying on models, e. g., E-model without paying careful attention towards the basic assumptions for which these models have been validated and developed. Conversely, quality engineering researchers treat network as a black-box which can be abstracted to few parameters that can be easily modelled to get quality metrics. While black-box approach is useful in general, however it is confusing due to the fact that multiple networking conditions may lead to the same quality outcomes.

Our efforts in this thesis are to work towards closing this gap and to leverage the strength of two different technical expertises. During our work, there are many instances when understanding the results obtained from our subjective tests were not possible before we augment it with the information from our network traces collected during the experiments. For instance, in one test we found that even with very good networking conditions, users continuously provide low ratings for video quality. We could only figure this out after analyzing network traces, which points out that due to large MTU, Mobile IP tunnel overhead force IP packet fragmentation. The small extra delay due to IP fragmentation renders video quality to suffer and users are able to perceive it quite well. Going back to black-box approaches, such phenomenas cannot be understood just from looking at quality dimension. We also find that unexpected results can come from the networking assumptions as well. For example, we expect to see high impact of network handovers, however it turns out that users get more annoyed with codec switching and bit-rate switching during the ongoing multi-media session. These are just few examples that highlight the importance of bringing up a consolidated view comprising of networking effects and quality engineering.

Besides different assumptions, we believe that another area where we can benefit more between networking and quality engineering is to develop tools such as VoIP and video clients that can be instrumented to get information about internal details such as jitter buffer. Not to forget, there are still multiple elements sitting between network layer and the user, for example, web-browsers, jitter buffers, etc. Similarly, we found that the major research from the networking community is derived based on considering only the large flows. However, our work accentuates the importance of short flows. These flows are important to understand as they define user experience, e. g., *Web-QoE*. While users are the ultimate judge of the services they get, as researchers our goal is to observe the reasons that lead to such judgements. All our efforts in this thesis are aimed to achieve this goal, however we have just scratched the surface and our work leads to many future directions.

9.3 Future directions

While Internet will continue to embrace new technologies and devices, we believe that the methodologies presented in this thesis for assessing network performance and user experiences can easily be extended to other environments. We would like to pursue the following research questions in the future.

- In Chapter 4, we analyze the impact of small buffers and high load on flows with regards to packet losses. We would like to extend this work to examine how much delay per-flow is induced by these losses and how it impacts end users perception. Delays in different segments of the Internet play a vital role in defining user perception. Recent studies of buffer bloat problem have highlighted the presence of the large buffers which act as to defeat the fundamental congestion control mechanism of TCP. We would like to extend our approach for wireless environments, especially LTE to examine how much new wireless technologies are impacted with the presence of large buffers.

- In Chapter 5, we propose flow-class approach to highlight the differences of QoS among flows of different applications. We point out the impact of congestion and highlight different network segments which are more problematic for applications such as HTTP and P2P. We endeavor to answer a controversial question that whom to blame among ISPs or users when faced with problems. We would like to explore more of these issues from home networking perspectives which is not only becoming complex but a hot research area. We would also like to compare our results with different networking environments such as Cable, FTTH, and LTE in different parts of the world to establish geographical comparison of Internet services. We are also interested in understanding Web-QoE in different networks. Another effort that we want to pursue is to model the flow performance of different applications according to our observations.

- Chapter 6, 7, and 8 present user QoE studies for few selected conditions. We believe, as the Internet continues to grow, more critical networking conditions will emerge. We would like to first identify prevalent conditions in new upcoming wireless technologies such as LTE and perform similar studies for VoIP, video, and mobile IPTV for broader range of conditions. We note that our quality studies for NGMNs are aimed at mobile devices having low resolution displays. As LTE promises high data rates and new mobile devices with higher resolution displays are available, therefore we would like to explore how such combination works in relation to user perception. We believe our cross-layer approach can prove useful to develop understanding of mobility management of users based on their experience.

Overall, better connections of QoE and network performance studies result in an improved modeling of user experience, e. g., Web-QoE.

10 Acknowledgments

This thesis would not have been possible without the guidance, encouragement, and support of numerous people and institutions around the world who contributed wholeheartedly to its success. I would like to collectively thank you each for your support from the bottom of my heart for making this such an enjoyable and memorable experience.

First, and foremost, I would like to express my deepest gratitude to my advisor Anja Feldmann for her invaluable guidance, gracious support, and technical advice. I know, I cannot thank her adequately. During my five and a half year stay at TU Berlin/T-Labs, she not only taught me advanced topics in networking but also trained me how to define and tackle networking research problems for my thesis. She showed me how to compete at the highest level in the research community by thinking deeply, being precise, and giving attention to the details. Her exceptional ability to quickly point out weaknesses of my work and constructive feedback have greatly helped me in improving the quality of my work. I am extremely grateful that she introduced me to the other people working on similar research topics in the research community and facilitated my research visits to UCLA, Stanford, Berkeley, and many conferences in different parts of the world. She taught me how to "aim high", "be persistent", and "be patient" at the same time to achieve success. Anja, I consider it my good fortune to have you as my advisor. Thank you!

Moreover, I am truly indebted and thankful to Steve Uhlig for his mentoring, support, guidance, candid feedback, encouragement and care. He always motivated me to strive for the best. Steve provided me great training at every step of the thesis, from problem formulation to evaluation and finally writing, by challenging and questioning the results with his excellent feedback. I am extremely thankful for his enormous time and wisdom despite his very busy schedule. Steve's passion for research and teaching has indeed a profound impact on me as an academic and will remain a constant source of inspiration for the rest of my career. Many discussions about research and life with Steve kept my spirits high during my difficult times of studies. Beside many things, he has instilled in me passion for reading books on various different topics. I have also benefited from his personal library. Steve, it is my pleasure and honor to be your *padawan*.

I would also like to express my gratitude to Walter Willinger. He invited me to intern with the multi-resolution research program at IPAM, UCLA California. During my multiple visits to UCLA I was able to expand my research vision by having insightful discussions with Walter Willinger, Paul Barford, Jon Doyle, Bruce Maggs and many other brilliant grad students and researchers from different countries. I am also thankful to Walter for accepting to be on my thesis committee and traveling from USA to be present in-person for my defense.

Next, I am thankful to the exciting intellectual environment of INET research group. There are not many places in the world where there is so much competence and brilliance under a single roof. Whenever I was stuck in some some problem, I could find some one at INET helping me out and widening my knowledge about possible solutions. I enjoyed many discussions during our mensa visits with all of them on various different topics. Again, I know it is not possible to name all people but I try to mention few of them.

I would like to express my sincere gratitude to Cigdem Sengul for her guidance, moral support, and advice during my studies. I admire how she magically turned raw results into an accepted paper in only three days' time. I would also like to thank Andreas Wundsam for his wonderful company, amazing support, and humor. We shared joys and sorrows of many accepted and rejected publications on our way to graduation. I admire his troubleshooting skills and his remarkable ability to convert any technically complex task into refreshing humor with his comments. I specially thank him for his company during our Stanford and Berkeley visits. I owe a special thanks to Bernhard Ager for his encouragement to learn, feedback on my work, and making me think deeply on my problems. He helped me at many times and in many ways. With his exceptional ways, he practically taught me how to partition a complex problem and nail it down. I strongly believe that his enthusiasm for Python programming made a strong influence on me and made me self-confident for my programing needs. I thoroughly enjoyed many discussions with him on research as well as on gliding and clouds. At the same time, I cannot forget many memorable kicker games that we played at T-labs. A special thanks goes to Nadi Sarrar for his friendship, support, collaboration, and feedback. I have enjoyed our time collaborating together in the hunt of happy flows. Much of the work in this thesis has benefited from the brainstorming sessions that we had during our collaboration.

Many thanks to Fabian Schneider, Gregor Maier and Jörg Wallerich for their guidance and sharing many tools; Wolfgang Mühlbauer and Dan Levin for their support in RouterLab course, I really enjoyed teaching with Wolfgang and Dan as they always have very good solutions for the technical challenges in the course; Harald and Doris Schiöberg for giving me a home-like feeling in Berlin, I will definitely remember our joint cooking sessions; Vinay Aggarwal for his good advice during my stay in Berlin; Gregor Schaffrath for boosting my moral at every stage of the thesis writing; Jan Böttger and Rainer May for their prompt technical support; Juhoon Kim for his time and determination for programing in Bro and fixing complex bugs; Oliver Hohlfeld, Ingmar Poese and Benjamin Frank for their help with the simulations and data; Thomas Hühn and Mustafa Al-Bado for great discussions. Last but not least I would like to acknowledge Britta Schneider and Birgit Hohmeier-Touré for university administration support.

I am also grateful to Prof. Sebastian Möller, Pablo Vidales, Blazej Lewcio, and Ina Wechsung from usability and quality group at TU-Berlin/T-labs for their support and collaboration during Mobisense project. I am grateful to Joel Sommers (Colgate University), Katrina LaCurts (MIT), JaeYong Yoo (GIST), and Shane Alcock (University of Waikato) for sharing their code and answering my numerous questions during the course of my Ph.D. work.

For the last five and half years I have been enjoying tremendous support from DAAD and Higher Education Commission (HEC) Pakistan. I would like to thank both DAAD and HEC, specially our DAAD coordinators Monika Osman and Kerstin Richter for helping me in bureaucratic and other matters. Many thanks for T-Labs management for providing such a wonderful research environment and facilitating visit of Dr. Leghari (Chairperson HEC) at T-labs. I would also like to thank Government of Pakistan for funding my Ph.D. studies.

Now, I would like to thank my parents, sisters, and my brother. Without their love, generous support, constant encouragement, care, and prayers this thesis would not have been possible. My father always provided me excellent education, advice, and encouragement. It was my mother's dream that I should pursue higher education abroad but at the same time she had to bear my absence with heavy-heart. My brother and sisters took care of my parents while I was away for my studies. I can hardly find words to express my gratitude for my family. Thanks for always being there when I needed you. This thesis is dedicated to you.

Finally, I would like to thank the almighty Allah for His mercy, kindness, blessings and giving me the strength and ability to complete this thesis. In the end, I humbly ask Him for further strength, courage, and perseverance to benefit others.

الحمد لله

Alhamdulillah

List of Figures

List of Tables

Bibliography

[1] VMware vSphere HypervisorTM (ESXi). http://www.vmware.com/products/vsphere-hypervisor/index.html.

[2] Beat the top 3 limiting constraints in 100G designs, Febraruy, 2011. http://www.eetimes.com/design/communications-design/4216552/Beat-the-top-3-limiting-constraints-in-100G-designs.

[3] BT inaugurates commercial FTTH offering. http://www.telegeography.com/products/commsupdate/articles/2011/11/16/bt-inaugurates-commercial-ftth-offering/.

[4] Carrier Intranet Service. http://www.telekom-icss.com/dtag/cms/contentblob/ICSS/en/330624/blobBinary/Carrier_Intranet_Service_Technical.pdf.

[5] CEO to shareholders: 50 billion connections 2020, April, 2010. http://www.ericsson.com/thecompany/press/releases/2010/04/1403231.

[6] Cisco CRS-3 1-Port 100 Gigabit Ethernet Interface Module. http://www.cisco.com/en/US/prod/collateral/routers/ps5763/CRS-1x100GE_DS.html.

[7] Elephants dream. http://www.elephantsdream.org/.

[8] Gartner, Press release, 2009. http://www.gartner.com/it/page.jsp?id=1170012.

[9] Google Fiber Blog, August 2011. http://googlefiberblog.blogspot.com/.

[10] IEEE 802.21. http://www.ieee802.org/21/index.html.

[11] IMS Research, Press release, August 2010. http://imsresearch.com/news-events/press-template.php?pr_id=1532.

[12] Internet World Stats, June 2011. http://www.internetworldstats.com/stats.htm.

[13] Juniper Networks, Press release, November, 2010. http://www.juniper.net/us/en/company/press-center/press-releases/2010/pr_2010_11_15-05_15.html.

[14] Managed Internet Service: Dedicated Access that Keeps You Connected. http://www.business.att.com/enterprise/Service/network-services/internet-connectivity/managed-internet-service/.

[15] NetFPGA Project. http://yuba.stanford.edu/NetFPGA/.

[16] NTT Communications provides high-quality Global IP Network Service with industry leading SLA. http://www.ntt.net/english/service/sla_ts.html.

[17] OECD Broadband Portal. http://www.oecd.org/sti/ict/broadband.

[18] Open Source SIP Stack and Media Stack for Presence, Instant Messaging, and Multimedia Communication. http://ww.pjsip.org/.

[19] Quavlive adaptive http streaming. http://www.quavlive.com/.

[20] Secgo Mobile IP Toolkit. http://www.secgo.com/.

[21] TCPDUMP. http://www.tcpdump.org/.

[22] Tcpreplay. http://tcpreplay.synfin.net/.

[23] Tribler. http://www.tribler.org/trac.

[24] TShark. http://www.wireshark.org/docs/man-pages/tshark.html.

[25] Verizon VoIP Voice Quality Service Level Agreement. http://www.verizonbusiness.com/terms/us/products/advantage/voicequality/.

[26] Youtube mobile gets a kick start. http://youtube-global.blogspot.com/2010/07/youtube-mobile-gets-kick-start.html.

[27] Definition of Quality of Experience. In *ITU-T: Appendix I to P.10/G.100* (2007). http://www.itu.int/md/T05-SG12-070116-TD-PLEN-0108/en.

[28] IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications, 2007.

[29] IEEE standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput, 2009.

[30] 3RD GENERATION PARTNERSHIP PROJECT. High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2, 3GPP TS 25.308. http://www.3gpp.org/ftp/Specs/html-info/25308.htm.

[31] ABDALLAH, A., MEDDOUR, D. E., AHMED, T., AND RASHEED, T. Cross-layer design for optimized video streaming over heterogeneous networks. In *Proceedings of the International Wireless Communications and Mobile Computing Conference* (June-July 2010).

[32] ABRY, P., AND VEITCH, D. Wavelet Analysis of Long-Range-Dependent Traffic. *IEEE Transactions on Information Theory 44*, 1 (1998), 2–15.

[33] AGARWAL, S., SINGH, J. P., MAVLANKAR, A., BACCICHET, P., AND GIROD, B. Performance of P2P Live Video Streaming Systems on a Controlled Test-bed. In *Proceedings of TRIDENTCOM* (2008).

[34] AGBOMA, F., AND LIOTTA, A. Addressing User Expectations in Mobile Content Delivery. *Mobile Information Systems 3*, 3,4 (December 2007), 153–164.

[35] AIKAT, J., KAUR, J., SMITH, F. D., AND JEFFAY, K. Variability in TCP round-trip times. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2003).

[36] AKAMAI. State of the internet, 4th quarter, 2011. Research Report http://www.akamai.com/stateoftheinternet/, April 2012.

[37] AKELLA, A., SESHAN, S., AND SHAIKH, A. An empirical evaluation of wide-area internet bottlenecks. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2003).

[38] ALCOCK, S., AND NELSON, R. Application flow control in YouTube video streams. *ACM SIGCOMM Computer Communication Review (CCR)* (2011).

[39] ALCOCK, S., AND NELSON, R. Passive detection of TCP congestion events. In *Proceedings of the International Conference on Telecommunications (ICT)* (2011).

[40] ALEXIOU, A., ANTONELLIS, D., AND BOURAS, C. Adaptive and reliable video transmission over UMTS for enhanced performance: Research Articles. *International Journal of Communication System 20* (January 2007), 65–81.

[41] ALLMAN, M., AVRACHENKOV, K., AYESTA, U., BLANTON, J., AND HURTIG, P. Early Retransmit for TCP and SCTP. RFC 5827, 2010.

[42] ALTMAN, E., AVRACHENKOV, K., AND BARAKAT, C. A stochastic model of TCP/IP with stationary random losses. *IEEE/ACM Transactions on Networking 13*, 2 (2005), 356–369.

[43] ANTONIADES, D., MARKATOS, E. P., AND DOVROLIS, C. One-click hosting services: a file-sharing hideout. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2009).

[44] APPENZELLER, G., KESLASSY, I., AND MCKEOWN, N. Sizing router buffers. In *Proceedings of the ACM SIGCOMM* (2004).

[45] AVRACHENKOV, K., AYESTA, U., BROWN, P., AND NYBERG, E. Differentiation between Short and Long TCP Flows: Predictability of the Response Time. In *Proceedings of the IEEE INFOCOM* (2004).

[46] BARHAM, P., DRAGOVIC, B., FRASER, K., HAND, S., HARRIS, T., HO, A., NEUGEBAUER, R., PRATT, I., AND WARFIELD, A. XEN and the art of virtualization. In *Proceedings of the ACM symposium on Operating systems principles* (2003).

[47] BEHESHTI, N., GANJALI, Y., GHOBADI, M., MCKEOWN, N., AND SALMON, G. Experimental study of router buffer sizing. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2008).

[48] BEHESHTI, N., GANJALI, Y., GHOBADI, M., MCKEOWN, N., AND SALMON, G. Time-sensitive network experiments. Tech. Rep. TR08-UT-SNL-04-30-00, April 2008.

[49] BLANTON, E., AND ALLMAN, M. On making TCP more robust to packet reordering. *ACM SIGCOMM Computer Communication Review (CCR)* (2002).

[50] BLANTON, E., ALLMAN, M., FALL, K., AND WANG, L. A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP. RFC 3517, 2003.

[51] BOHACEK, S., HESPANHA, P., LEE, J., LIM, C., AND OBRACZKA, K. A new TCP for persistent packet reordering. *IEEE/ACM Transactions on Networking* (2006).

[52] BOLOT, J.-C. End-to-end packet delay and loss behavior in the Internet. In *Proceedings of the ACM SIGCOMM* (1993).

[53] BORGNAT, P., DEWAELE, G., FUKUDA, K., ABRY, P., AND CHO, K. Seven years and one day: Sketching the evolution of internet traffic. In *Proceedings of the IEEE INFOCOM* (2009).

[54] BUSH, R., GRIFFIN, T., MAO, Z., PURPUS, E., AND STUTSBACH, D. Happy packets: Some initial results. In *Proceedings of the IEEE INFOCOM* (2004).

[55] BUT, J., KELLER, U., AND ARMITAGE, G. Passive TCP Stream Estimation of RTT and Jitter Parameters. In *Proceedings of the IEEE Conference on Local Computer Networks* (2005).

[56] BUTKIEWICZ, M., MADHYASTHA, H., AND SEKAR, V. Understanding website complexity: measurements, metrics, and implications. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2011).

[57] CALVAGNA, A., AND MODICA, G. D. A user-centric analysis of vertical handovers. In *Proceedings of the ACM international workshop on Wireless mobile applications and services on WLAN hotspots* (2004).

[58] CAO, J., CLEVELAND, W., LIN, D., AND SUN, D. On the nonstationarity of Internet traffic. In *Proceedings of the ACM SIGMETRICS* (2001).

[59] CARSON, M., AND SANTAY, D. NIST Net: a Linux-based network emulation tool. *ACM SIGCOMM Computer Communication Review (CCR) 33*, 3 (2003), 111–126.

[60] CASETTI, C., GERLA, M., MASCOLO, S., SANADIDI, M., AND WANG, R. TCP Westwood: End-to-End congestion control for wired/wireless networks. *Wireless Networks* (2002).

[61] CHAN, M. C., AND RAMJEE, R. TCP/IP performance over 3G wireless links with rate and delay variation. In *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom)* (September 2002).

[62] Cisco Visual Networking Index: Global Mobile Data Traffic. `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html`.

[63] Cisco visual networking index: Forecast and methodology, 2010-2015. `http://www.cisco.com/`.

[64] CROVELLA, M. E., AND BESTAVROS, A. Self-similarity in World Wide Web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking* (1997).

[65] DAHLMAN, E., PARKVALL, S., SKOLD, J., AND BEMING, P. *3G Evolution: HSPA and LTE for Mobile Broadband*. Academic Press, 2008.

[66] DESHPANDE, P., HU, X., AND DAS, S. R. Performance comparison of 3G and metro-scale WiFi for vehicular network access. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (Nov. 2010).

[67] DREGER, H., FELDMANN, A., MAI, M., PAXSON, V., AND SOMMER, R. Dynamic application-layer protocol analysis for network intrusion detection. In *Proceedings of the USENIX Security Symposium* (2006).

[68] DREGER, H., FELDMANN, A., PAXSON, V., AND SOMMER, R. Operational experiences with high-volume network intrusion detection. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)* (2004).

[69] DUKKIPATI, N., MATHIS, M., CHENG, Y., AND GHOBADI, M. Proportional rate reduction for TCP. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2011).

[70] DUKKIPATI, N., REFICE, T., CHENG, Y., CHU, J., HERBERT, T., AGARWAL, A., JAIN, A., AND SUTIN, N. An argument for increasing TCP's initial congestion window. *ACM SIGCOMM Computer Communication Review (CCR)* (2010).

[71] DURAN, A., DEL PLIEGO, E., AND ALONSO, J. Effects of handover on voice quality in wireless convergent networks. In *Proceedings of the IEEE Radio and Wireless Symposium* (January 2007).

[72] ENACHESCU, M., GOEL, A., ROUGHGARDEN, T., GANJALI, Y., AND MCKEOWN, N. Routers with very small buffers. In *Proceedings of the IEEE INFOCOM* (2006).

[73] ENNAJI, Y., BOULMALF, M., AND ALAOUI, C. Experimental analysis of video performance over wireless local area networks. In *Proceedings of the International Conference on Multimedia Computing and Systems* (April 2009), pp. 488 –494.

[74] FELDMANN, A., GILBERT, A. C., WILLINGER, W., AND KURTZ, T. G. The changing nature of network traffic: Scaling phenomena. *ACM SIGCOMM Computer Communication Review (CCR) 28*, 2 (1998).

[75] FRALEIGH, C., MOON, S., LYLES, B., COTTON, C., KHAN, M., MOLL, D., ROCKELL, R., SEELY, T., AND DIOT, C. Packet-Level Traffic Measurements from the Sprint IP Backbone. *IEEE/ACM Transactions on Networking 17* (2003), 6–16.

[76] GASS, R., AND DIOT, C. An experimental performance comparison of 3G and Wi-Fi. In *Proceedings of the Passive and Active Measurement Conference (PAM)* (May 2010).

[77] GERBER, A., PANG, J., SPATSCHECK, O., AND VENKATARAMAN, S. Speed testing without speed tests: estimating achievable download speed from passive measurements. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2010).

[78] GOJMERAC, I., HAMMER, F., RICCIATO, F., TRAN, H., AND ZIEGLER, T. Scalable QoS: state-of-the-art architectural solutions and developments. Tech. rep., 2004.

[79] GOUDARZI, M., SUN, L., AND IFEACHOR, E. PESQ and 3SQM measurement of voice quality over live 3G networks. In *Proceedings of the International Conference MESAQIN* (June 2009).

[80] GRÖNVALL, B., AND MARSH, I. Performance evaluation of voice handovers in real 802.11 networks. In *Proceedings of the second International workshop on wireless network measurement* (2006).

[81] GUDE, N., KOPONEN, T., PETTIT, J., PFAFF, B., CASADO, M., MCKEOWN, N., AND SHENKER, S. NOX: towards an operating system for networks. In *ACM SIGCOMM Computer Communication Review (CCR)* (July 2008), vol. 38.

[82] HA, S., RHEE, I., AND XU, L. CUBIC: A new TCP-friendly high-speed TCP variant. *ACM SIGOPS Operating System Review* (2008).

[83] HAFSAOUI, A., COLLANGE, D., AND URVOY-KELLER, G. Revisiting the Performance of Short TCP Transfers. In *Proceedings of the International IFIP-TC 6 Networking Conference* (2009).

[84] HALEPOVIC, E., WU, Q., WILLIAMSON, C., AND GHADERI, M. TCP over WiMAX: A measurement study. In *Proceedings of MASCOTS* (September 2008).

[85] HASSENZAHL, M. User experience (UX): towards an experiential perspective on product quality. In *Proceedings of the 20th International Conference of the Association Francophone d'Interaction Homme-Machine* (2008), pp. 11–15.

[86] HASSENZAHL, M., AND TRACTINSKY, N. User experience - a research agenda. *Behaviour & Information Technology 25*, 2 (2006), 91–97.

[87] HURTIG, P., JOHN, W., AND BRUNSTROM, A. Recent Trends in TCP Packet-Level Characteristics. In *Proceedings of the International Conference on Networking and Services (ICNS)* (2011).

[88] HUSZAK, A., AND IMRE, S. Analysing GOP structure and packet loss effects on error propagation in MPEG-4 video streams. In *Proceedings of the International Symposium on Communications, Control and Signal Processing (ISCCSP)* (March 2010), pp. 1–5.

[89] IHM, S., AND PAI, V. S. Towards understanding modern web traffic. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2011).

[90] ITU G.992.1-5. *ADSL standards*. ITU.

[91] ITU-T REC. G.107. *- Amendment 1: "New Appendix II - Provisional impairment factor framework for wideband speech transmission"*. ITU, 2006.

[92] ITU-T REC. G.107. *The E-model, a Computational Model for Use in Transmission Planning*. ITU, 2009.

[93] ITU-T REC. G.1070. *Opinion Model for Video-Telephony Applications*. International Telecommunication Union, april 2007.

[94] ITU-T REC. G.711. *Pulse Code Modulation (PCM) of Voice Frequencies*. ITU, 1988.

[95] ITU-T REC. G.723.1. *Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s*. ITU, 1996.

[96] ITU-T REC. G.729. *Coding of Speech at 8kbit/s Using Conjugate-structure Algebraic-Code Excited Linear-Prediction (CS-ACELP)*. ITU, 1996.

[97] ITU-T REC. J.144. *Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference*. International Telecommunication Union, February 2004.

[98] ITU-T REC. J.341. *Objective perceptual multimedia video quality measurement of HDTV for digital cable television in the presence of a full reference*. International Telecommunication Union, January 2011.

[99] ITU-T REC. P.800. *Methods of Subjective Determination of Transmission Quality*. International Telecommunication Union, 1996.

[100] ITU-T REC. P.862. *Perceptual Evaluation of Speech Quality(PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs*. ITU, 2001.

[101] ITU-T REC. P.862.2. *Wideband Extension to Recommendation P.862 for the Assessment of Wideband Telephone Networks and Speech Codecs*. ITU, 2005.

[102] ITU-T REC. P.910. *Subjective video quality assessment methods for multimedia applications*. International Telecommunication Union, December 1998.

[103] IZAL, M., URVOY-KELLER, G., BIERSACK, E., FELBER, P., HAMRA, A., AND GARCÉS-ERICE, L. Dissecting bittorrent: Five months in a torrent's lifetime. In *Proceedings of the Passive and Active Measurement Conference (PAM)* (2004).

[104] JACOBSON, V. Modified TCP congestion control algorithm. End-to-end interest mailing list, April 1990 . ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt.

[105] JACOBSON, V., BRADEN, R., AND BORMAN, D. TCP Extensions for High Performance. RFC 1323, 1992.

[106] JAIN, R. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, Inc., 1991.

[107] JAISWAL, S., IANNACCONE, G., DIOT, C., KUROSE, J., AND TOWSLEY, D. Measurement and classification of out-of-sequence packets in a tier-1 IP backbone. *IEEE/ACM ToN* (2007).

[108] JEKOSCH, U. *Voice and Speech Quality Perception: Assessment and Evaluation (Signals and Communication Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[109] JIANG, H., AND DOVROLIS, C. Passive estimation of TCP round-trip times. *ACM SIGCOMM Computer Communication Review (CCR)* (2002).

[110] JIANG, H., AND DOVROLIS, C. Why is the Internet traffic bursty in short time scales? In *Proceedings of the ACM SIGMETRICS* (2005), pp. 241–252.

[111] JOUMBLATT, D., TEIXEIRA, R., CHANDRASHEKAR, J., AND TAFT, N. Performance of networked applications: the challenges in capturing the user's perception. In *Proceedings of the ACM SIGCOMM workshop W-MUST* (2011), pp. 37–42.

[112] JURVANSUU, M., PROKKOLA, J., HANSKI, M., AND PERÄLÄ, P. HSDPA Performance in Live Networks. In *Proceedings of the International Conference on Communications (ICC)* (2007).

[113] KARAGIANNIS, T., MOLLE, M., FALOUTSOS, M., AND BROIDO, A. A nonstationary Poisson view of Internet traffic. In *Proceedings of the IEEE INFOCOM* (2004).

[114] KARAGIANNIS, T., MOLLE, M., FALOUTSOS, M., AND BROIDO, A. A nonstationary poisson view of internet traffic. In *Proceedings of the IEEE INFOCOM* (2004).

[115] KHAN, A., SUN, L., JAMMEH, E., AND IFEACHOR, E. Quality of experience-driven adaptation scheme for video applications over wireless networks. *IET Communications 4*, 11 (23 2010), 1337 –1347.

[116] KIVITY, A., KAMAY, Y., LAOR, D., LUBLIN, U., AND LIGUORI, A. KVM: the Linux Virtual Machine Monitor. In *Proceedings of the Linux Symposium* (2007).

[117] LABOVITZ, C., IEKEL-JOHNSON, S., MCPHERSON, D., OBERHEIDE, J., AND JAHANIAN, F. Internet inter-domain traffic. In *Proceedings of the ACM SIGCOMM* (2010).

[118] LAR, S., AND LIAO, X. An initiative for a classified bibliography on TCP/IP congestion control. *Journal of Network and Computer Applications* (2012).

[119] LEE, C., LEE, D. K., YI, Y., AND MOON, S. Operating a network link at 100%. In *Proceedings of the Passive and Active Measurement Conference (PAM)* (2011), pp. 1–10.

[120] LEICHTMAN RESEARCH GROUP, INC. Smart TV Sets – Today's Shimmer? Research Notes http://www.leichtmanresearch.com/research/notes04_2012.pdf, 2012.

[121] LEIGHTON, T. Improving performance on the internet. *JACM* (2009).

[122] LELAND, W. E., TAQQU, M. S., WILLINGER, W., AND WILSON, D. V. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking 2* (February 1994), 1–15.

[123] LEVIN, D., WUNDSAM, A., MEHMOOD, A., AND FELDMANN, A. BERLIN: The Berlin Experimental Router Laboratory for Innovative Networking. In *Proceedings of TRIDENTCOM* (May 2010).

[124] LEWCIO, B., BELMUDEZ, B., MEHMOOD, A., WÄLTERMANN, M., AND MÖLLER, S. Video Quality in Next Generation Mobile Networks - Perception of Time-Varying Transmission. In *Proceedings of IEEE International Communications Quality and Reliability Workshop (CQR)* (May 2011).

[125] LEWCIO, B., WÄLTERMANN, M., MÖLLER, S., AND VIDALES, P. E-model supported switching between narrowband and wideband speech quality. In *Proceedings of the International Workshop on Quality of Multimedia Experience (QoMEX)* (2009).

[126] LOISEAU, PATRICK AND GONÇALVES, PAULO AND DEWAELE, GUILLAUME AND BORGNAT, PIERRE AND ABRY, PATRICE AND PRIMET, PASCALE VICAT-BLANC. Investigating self-similarity and heavy-tailed distributions on a large-scale experimental facility. *IEEE/ACM Transactions on Networking 18* (August 2010), 1261–1274.

[127] MAIER, G., FELDMANN, A., PAXSON, V., AND ALLMAN, M. On dominant characteristics of residential broadband internet traffic. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2009).

[128] MAIER, G., SCHNEIDER, F., AND FELDMANN, A. A first look at mobile hand-held device traffic. In *Proceedings of the Passive and Active Measurement Conference (PAM)* (April 2010), pp. 161–170.

[129] MARPE, D., WIEGAND, T., AND SULLIVAN, G. The h.264/mpeg4 advanced video coding standard and its applications. *Communications Magazine, IEEE 44*, 8 (August 2006), 134 –143.

[130] MARTIN, J., AND NILSSON, A. On service level agreements for IP networks. In *Proceedings of the IEEE INFOCOM* (2002), vol. 2, pp. 855 – 863 vol.2.

[131] MAWI Working Group Traffic Archive. http://mawi.wide.ad.jp/mawi.

[132] MCKEOWN, N., ANDERSON, T., BALAKRISHNAN, H., PARULKAR, G., PETERSON, L., REXFORD, J., SHENKER, S., AND TURNER, J. OpenFlow: Enabling innovation in campus networks. In *ACM SIGCOMM Computer Communication Review (CCR)* (2008), vol. 38.

[133] MEHMOOD, A., AGARWAL, S., SENGUL, C., AND FELDMANN, A. Mobile Video QoE in Future Mobile Communications. In *Proceedings of the Passive and Active Measurement Conference (PAM)* (April 2010).

[134] MEHMOOD, A., HOHLFELD, O., LEVIN, D., WUNDSAM, A., CIUCU, F., SCHNEIDER, F., FELDMANN, A., AND BRAUN, R.-P. The Routerlab – Emulating Internet Characteristics in a Room. In *Proceedings of 11th ITG Symposium on Photonic Networks (11. ITG-Fachtagung Photonische Netze)* (May 2010).

[135] MEHMOOD, A., LEWCIO, B., VIDALES, P., FELDMANN, A., AND MÖLLER, S. Understanding Signal-Based Speech Quality Prediction in Future Mobile Communications. In *Proceedings of IEEE International Conference on Communications (ICC)* (May 2010).

[136] MEHMOOD, A., SARRAR, N., UHLIG, S., AND FELDMANN, A. How happy are your flows: an empirical study of packet losses in router buffers. Tech. Rep. 2012-07, ISSN:1436-9915, May 2012.

[137] MEHMOOD, A., SENGUL, C., SARRAR, N., AND FELDMANN, A. Understanding Cross-Layer Effects on Quality of Experience for Video over NGMN. In *Proceedings of IEEE International Conference on Communications (ICC)* (June 2011).

[138] MEHMOOD, A., WUNDSAM, A., UHLIG, S., LEVIN, D., SARRAR, N., AND FELDMANN, A. QoE-Lab: Towards evaluating Quality of Experience for Future Internet Conditions. In *Proceedings of TRIDENTCOM* (April 2011).

[139] MELLIA, M., LO CIGNO, R., AND NERI, F. Measuring IP and TCP behavior on edge nodes with Tstat. *Computer Networks 47*, 1 (Jan. 2005), 1–21.

[140] MELLIA, M., MEO, M., MUSCARIELLO, L., AND ROSSI, D. Passive analysis of TCP anomalies. *Computer Networks 52*, 14 (Oct. 2008), 2663–2676.

[141] MÖLLER, S., RAAKE, A., KITAWAKI, N., TAKAHASHI, A., AND WÄLTERMANN, M. Impairment Factor Framework for Wideband Speech Codecs. *IEEE Transactions on Audio, Speech, and Language Processing 14*, 6 (2006), 1969–1976.

[142] MOLLER, S., WALTERMANN, M., LEWCIO, B., KIRSCHNICK, N., AND VIDALES, P. Speech Quality While Roaming in Next Generation Networks. In *Proceedings of the International Conference on Communications (ICC)* (June 2009), pp. 1 –5.

[143] NGMN ALLIANCE. *Next Generation Mobile Networks Beyond HSPA & EVDO*, Dec. 2006. White paper, `http://www.ngmn.org/`.

[144] NGUYEN, H., AND ROUGHAN, M. On the correlation of internet packet losses. In *Proceedings of ATNAC* (2008).

[145] NGUYEN, H., AND THIRAN, P. Network loss inference with second order statistics of end-to-end flows. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2007).

[146] NORTON, W. B. *Video Internet: The Next Wave of Massive Disruption to the US Peering.*, September 2006. White paper, `http://drpeering.net/white-papers/Video-Internet-The-Next-Wave-Of-Massive-Disruption-To-The-U.S.-Peering-Ecosystem.html`.

[147] OF PRINCETON UNIVERSITY, T. T. PlanetLab. `http://www.planet-lab.org/`.

[148] OPENEPC. Always Best Connected - Access Network Discovery and Selection. `http://www.openepc.net/en/openepc/demo_scenarios/`.

[149] P. COVERDALE. *Definition of Quality of Experience*. International Telecommunication Union, Source: Nortel Networks, Canada, Nortel Networks, Canada 2003.

[150] PADHYE, J., FIROIU, V., TOWSLEY, D., AND KUROSE, J. Modeling TCP throughput: a simple model and its empirical validation. In *Proceedings of the ACM SIGCOMM* (1998).

[151] PANTOS, R., AND MAY, W. Http live streaming. `http://tools.ietf.org/html/draft-pantos-http-live-streaming-04`, June 2010.

[152] PARK, K., AND WILLINGER, W., Eds. *Self-Similar Network Traffic and Performance Evaluation*. Wiley-Interscience, 2000.

[153] PAXSON, V. End-to-end Internet packet dynamics. In *ACM SIGCOMM* (1997), pp. 139–152.

[154] PAXSON, V. *Measurements and analysis of end-to-end Internet dynamics*. PhD thesis, 1998.

[155] PAXSON, V. Bro: a system for detecting network intruders in real-time. *Computer Networks 31* (1999), 23–24.

[156] PAXSON, V., ALLMAN, M., H.K.CHU, AND SARGENT, M. Computing TCP's Retransmission Timer. RFC 2988, 2000.

[157] PAXSON, V., AND FLOYD, S. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking 3* (June 1995), 226–244.

[158] PENTIKOUSIS, K., BADR, H., AND ANDRADE, A. A comparative study of aggregate TCP retransmission rates. `http://arxiv.org/abs/1112.2292`.

[159] PEUHKURI, M. IP Quality of Service. Helsinki University of Technology, Laboratory of Telecommunications Technology, 1999. `http://www.netlab.tkk.fi/~puhuri/htyo/Tik-110.551/iwork/iwork.html`.

[160] PIAMRAT, K., VIHO, C., BONNIN, J.-M., AND KSENTINI, A. Quality of Experience Measurements for Video Streaming over Wireless Networks. In *Proceedings of ITNG* (April 2009), pp. 1184 –1189.

[161] PLISSONNEAU, L., COSTEUX, J.-L., AND BROWN, P. Analysis of peer-to-peer traffic on adsl. In *Proceedings of the Passive and Active Measurement Conference (PAM)* (2005), vol. 3431, pp. 69–82.

[162] POUWELSE, J., GARBACKI, P., EPEMA, D., AND SIPS, H. The bittorrent P2P file-sharing system: Measurements and analysis. In *Proceedings of IPTPS* (2005).

[163] PRASAD, R., AND DOVROLIS, C. Measuring the congestion responsiveness of Internet traffic. In *Proceedings of the Passive and Active Measurement Conference (PAM)* (2007).

[164] PYUN, J.-Y. Context-Aware Streaming Video System for Vertical Handover over Wireless Overlay Network. *IEEE Transactions on Consumer Electronics 54*, 1 (February 2008), 71 –79.

[165] QIAN, F., GERBER, A., MAO, M., SEN, S., SPATSCHECK, O., AND WILLINGER, W. TCP revisited: A fresh look at TCP in the wild. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2009).

[166] QUAN, L., AND HEIDEMANN, J. On the characteristics and reasons of long-lived internet flows. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2010).

[167] RAAKE, A. *Speech Quality of VoIP: Assessment and Prediction*. John Wiley & Sons, Ltd, September 2006.

[168] RAAKE, A., GARCIA, M.-N., MÖLLER, S., BERGER, J., KLING, F., LIST, P., JOHANN, J., AND HEIDEMANN, C. T-V-Model: Parameter-based Prediction of IPTV Quality. In *Proceedings of the IEEE ICASSP '08* (2008).

[169] REWASKAR, S., KAUR, J., AND SMITH, F. A passive state-machine approach for accurate analysis of TCP out-of-sequence segments. *ACM SIGCOMM Computer Communication Review (CCR)* (2006).

[170] RIZZO, L. Dummynet: a simple approach to the evaluation of network protocols. *ACM SIGCOMM Computer Communication Review (CCR)* (1997).

[171] ROSENBERG, J., AND SCHULZRINNE, H. An Offer/Answer Model with Session Description Protocol (SDP). RFC 3264 (Proposed Standard), June 2002.

[172] SANDVINE INTELLIGENT BROADBAND NETWORKS. Global broadband phenomena report. Research Report `http://www.sandvine.com/news/global_broadband_trends.asp`, 2012.

[173] SAXENA, M., SHARAN, U., AND FAHMY, S. Analyzing Video Services in Web 2.0: A Global Perspective. In *Proceedings of the ACM Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)* (May 2008).

[174] SCHORR, A., KASSLER, A., AND PETROVIC, G. Adaptive media streaming in heterogeneous wireless networks. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing* (September 2004), pp. 506–509.

[175] SCHULZE, H., AND MOCHALSKI, K. Internet study 2007. `http://www.ipoque.com/resources/internet-studies/` (need to register), 2007.

[176] SCHULZE, H., AND MOCHALSKI, K. Internet study 2008/2009. `http://www.ipoque.com/resources/internet-studies/` (need to register), 2009.

[177] SCHULZRINNE, H., AND WEDLUND, E. Application-layer mobility using SIP. *ACM SIGMOBILE Mobile Computing and Communications Review 4* (July 2000), 47–57.

[178] SEN, S., AND WANG, J. Analyzing peer-to-peer traffic across large networks. *IEEE/ACM Transactions on Networking 12*, 2 (Apr. 2004), 219–232.

[179] SHAIKH, A., REXFORD, J., AND SHIN, G. Load-sensitive routing of long-lived IP flows. In *Proceedings of the ACM SIGCOMM* (1999).

[180] SHIRAN, N., AND SHALLOM, I. Enhanced PESQ Algorithm for Objective Assessment of Speech Quality at a Continuous Varying Delay. In *Proceedings of the International Workshop on Quality of Multimedia Experience (QoMEX)* (2009).

[181] SIEKKINEN, M., URVOY-KELLER, G., BIERSACK, E., AND COLLANGE, D. A root cause analysis toolkit for TCP. *Computer Networks 52*, 9 (June 2008), 1846–1858.

[182] SOLDANI, D. Bridging QoE and QoS for Mobile Broadband Networks. In *Proceedings of the ETSI workshop on QoS, QoE and User Experience focusing on speech, multimedia conference tools.* (2010).

[183] SOMMERS, J. *Calibrated Network Measurement.* PhD thesis, Computer Sciences Department, University of Wisconsin-Madison, July 2007.

[184] SOMMERS, J., AND BARFORD, P. Self-configuring network traffic generation. In *Proceedings of the ACM Internet Measurement Conference (IMC)* (2004).

[185] SOMMERS, J., BARFORD, P., DUFFIELD, N., AND RON, A. Accurate and efficient SLA compliance monitoring. In *Proceedings of ACM SIGCOMM* (2007).

[186] Strangeloop Site Optimizer. `http://www.strangeloopnetworks.com/assets/Uploads/SO-Datasheet.pdf`.

[187] SUNDARESAN, S., DE DONATO, W., FEAMSTER, N., TEIXEIRA, R., CRAWFORD, S., AND PESCAPÈ, A. Broadband internet performance: a view from the gateway. In *Proceedings of the ACM SIGCOMM* (2011).

[188] TAKAHASHI, A., HANDS, D., AND BARRIAC, V. Standardization activities in the ITU for a QoE assessment of IPTV. *Communications Magazine, IEEE 46*, 2 (february 2008), 78 –84.

[189] TAN, K., SONG, J., ZHANG, Q., AND SRIDHARAN, M. A Compound TCP Approach for High-Speed and Long Distance Networks. In *Proceedings of the IEEE INFOCOM* (2006).

[190] UHLIG, S. Non-stationarity and high-order scaling in TCP flow arrivals: a methodological analysis. *ACM SIGCOMM Computer Communication Review (CCR) 34*, 2 (2004).

[191] VACIRCA, F., ZIEGLER, T., AND HASENLEITHNER, E. An algorithm to detect TCP spurious timeouts and its application to operational UMTS/GPRS networks. *Computer Networks 50*, 16 (Nov. 2006).

[192] VIDALES, P., KIRSCHNICK, N., STEUER, F., LEWCIO, B., WÄLTERMANN, M., AND MÖLLER, S. Mobisense Testbed: Merging User Perception and Network Performance. In *Proceedings of TRIDENTCOM* (2008).

[193] VORAN, S. A Basic Experiment on Time-Varying Speech Quality. In *Proceedings of the International Conference MESAQIN* (June 2005).

[194] WALTERMANN, M., LEWCIO, B., VIDALES, P., AND MOLLER, S. A technique for seamless voip-codec switching in next generation networks. In *Proceedings of the International Conference on Communications (ICC)* (May 2008), pp. 1772 –1776.

[195] WÄLTERMANN, M., SCHOLZ, K., RAAKE, A., HEUTE, U., AND MÖLLER, S. Underlying quality dimensions of modern telephone connections. In *Proceedings of INTERSPEECH* (2006).

[196] WANG, Y., LU, G., AND LI, X. A Study of Internet Packet Reordering. In *Proceedings of ICOIN* (2004).

[197] WANG, Z., LU, L., AND BOVIK, A. Video quality assessment using structural distortion measurement. In *Proceedings of International Conference on Image Processing* (2002), vol. 3.

[198] WIEGAND, T., SCHWARZ, H., JOCH, A., KOSSENTINI, F., AND SULLIVAN, G. Rate-constrained coder control and comparison of video coding standards. *IEEE Transactions on circuits and systems for video technology 13*, 7 (july 2003), 688 – 703.

[199] WILLINGER, W., TAQQU, M., SHERMAN, R., AND WILSON, D. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking 5*, 1 (feb 1997), 71 –86.

[200] WINKLER, S., AND MOHANDAS, P. The evolution of video quality measurement: From PSNR to hybrid metrics. *IEEE Transaction on Broadcasting 54*, 3 (September 2008), 660–668.

[201] WUNDSAM, A., MEHMOOD, A., FELDMANN, A., AND MAENNEL, O. Network Troubleshooting with Shadow VNets. In *SIGCOMM '09: Demo Session of the 2009 conference on Applications, technologies, architectures, and protocols for computer communications* (August 2009).

[202] WUNDSAM, A., MEHMOOD, A., FELDMANN, A., AND MAENNEL, O. Network Troubleshooting with Mirror VNets. In *Proceedings of IEEE Globecom 2010 Workshop of Network of the Future (FutureNet-III)* (December 2010).

[203] YAJNIK, M., MOON, S., KUROSE, J., AND TOWSLEY, D. Measurement and modelling of the temporal dependence in packet loss. In *Proceedings of the IEEE INFOCOM* (1999).

[204] YOO, J., HUHN, T., AND KIM, J. Active capture of wireless traces: overcome the lack in protocol analysis. In *Proceedings of WiNTECH* (2008).

[205] ZHAI, G., CAI, J., LIN, W., YANG, X., AND ZHANG, W. Three Dimensional Scalable Video Adaptation via User-End Perceptual Quality Assessment. *IEEE Transactions on Broadcasting 54*, 3 (sept. 2008), 719 –727.

[206] ZHANG, Y., BRESLAU, L., PAXSON, V., AND SHENKER, S. On the characteristics and origins of internet flow rates. In *Proceedings of the ACM SIGCOMM* (2002).