



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

Impact of Buffering on Quality of Experience

vorgelegt von
Oliver Hohlfeld (M.Sc.)
aus Langen

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

DOKTOR DER NATURWISSENSCHAFTEN
- DR. RER. NAT. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. Sebastian Möller, TU Berlin
Gutachterin: Prof. Anja Feldmann, Ph.D., TU Berlin
Gutachter: Prof. Paul Barford, Ph.D., University of Wisconsin—Madison
Gutachter: Prof. Dr.-Ing. Alexander Raake, TU Berlin

Tag der wissenschaftlichen Aussprache: 28. Oktober 2013

Berlin 2014
D 83

Eidesstattliche Erklärung

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

Datum Oliver Hohlfeld (M.Sc.)

Abstract

The Internet has become an essential part of the lives of millions of people and an invaluable asset to businesses. As an emerging trend, data storage and processing is shifting to the Cloud (e.g., Google Apps, or Cloud gaming), making users more and more dependent on the network to perform their daily activities. Despite the crucial importance of Internet services, they remain susceptible to bad service quality. One particular factor influencing service quality is buffering at various layers.

This thesis assess the impact of buffering on Quality of Experience (QoE). QoE is an active research area aiming to quantify the users' perception of Internet services. This is challenging since the users' perception is subjective. This thesis tackles this challenge by using a multi-disciplinary approach that combines QoE and networking research to take a cross-layer perspective on network and application buffering.

Network buffering occurs in hosts, switches, and routers throughout the Internet. It impacts network performance by contributing delays, jitter, and packet losses. Loss-based degradations of video quality are illustrated in a first evaluation. Motivated by this observation, Scalable Video Coding is discussed to optimize video QoE in phases of congestion. An evaluation of SVC dimensions shows that spatial scalability yields better QoE scores than temporal scalability. Further, QoE impacts of model based packet loss generators—e.g., as used in QoE studies—are evaluated. It is shown that the model choice impacts quality indicators, thus model choice matters.

The size of network buffers influences network performance by controlling the level of introduced delay, jitter, and packet loss. The choice of 'proper' buffer sizing guidelines remains an unresolved and controversially discussed topic since decades. In this context, this thesis presents the first comprehensive study on the impact of buffer sizes on *Quality of Experience*, involving relevant user applications (e.g., voice, video, and web browsing), real hardware, and realistic workload. While bloated buffers *can* degrade QoE, buffer sizes that follow standard sizing guidelines significantly impact QoS metrics, but impact QoE metrics only marginally. Limiting congestion, may thus thus yield more immediate QoE improvements than optimal buffer sizes.

Application buffering is used to compensate for performance variations, e.g., originating from network buffering. One example is the buffer-based, proprietary retransmission scheme in a major IPTV system. This thesis provides insights into the functioning of this scheme and motivates the extension of QoE metrics to account for client-side error recovery to prevent QoE mispredictions. To optimize Web QoE, a hit rate analysis of caching schemes is performed by focusing on YouTube video popularities. The thesis contributes an optimized caching scheme that offers higher cache hit rates than traditional Least Recently Used caches.

Finally, it broadens the view of QoE by discussing spam as major QoE determinant in e-mail. A large-scale study conducted over 3.5 years reveals insights into address harvesting as the origin of spam and proposes mechanisms for spam mitigation that can help to improve e-mail QoE.

Zusammenfassung

Das Internet hat sich zu einem elementaren Bestandteil im Leben von Millionen von Menschen und Unternehmen entwickelt. Im Zuge dieser Entwicklung verschiebt sich die Datenverarbeitung und -speicherung zunehmend in die Cloud (z.B. Google Apps oder Cloud-Spiele), was die Abhängigkeit vom Internet erhöht. Trotz der elementaren Bedeutung des Internets sind dessen Dienste anfällig für Dienstqualitätsprobleme. Ein Einflussfaktor dabei sind Puffer auf verschiedenen Protokollebenen.

Diese Dissertation untersucht den Einfluss dieser Puffer auf die Nutzerzufriedenheit (Quality of Experience, QoE). Dies gestaltet sich als Herausforderung, da die Zufriedenheit ein subjektives Maß ist. Diese Dissertation verfolgt daher einen interdisziplinären Ansatz, der QoE- und Netzwerkforschung zur Untersuchung von Puffern auf der Netzwerk- und Anwendungsebene kombiniert.

Auf der Netzwerkebene finden sich Internetweit Puffer in Hosts, Switches und Routern. Sie können die Dienstgüte durch Verzögerungen, Jitter und Paketverluste beeinflussen. Eine erste Untersuchung illustriert die negative Auswirkungen solcher Verluste auf die Video QoE. Qualitätsverbesserungen können hierbei durch den Einsatz von Scalable Video Coding erzielt werden. Eine Untersuchung der SVC-Skalierungsdimensionen zeigt, dass spatial scalability zu besseren QoE Vorhersagen führt, als temporal scalability. Eine weitere Studie untersucht den QoE-Einfluss modellbasierter Paketverlustgeneratoren, die beispielsweise in QoE-Studien Verwendung finden. Es wird gezeigt, dass die Modellwahl die QoE-Ergebnisse beeinflusst.

Die Puffergröße beeinflusst die entstehende Verzögerung, Jitter und die Paketverluste. Trotz jahrzentaler Forschung und operativer Erfahrung wird die "richtige" Pufferdimensionierung kontrovers diskutiert. Diese Arbeit präsentiert die erste umfassende Studie über den Einfluss der Pufferdimensionierung auf die QoE von Internetanwendungen wie Telefonie, Videostreaming und Webbrowsing. Während überdimensionierte Puffer die QoE beeinträchtigen *können*, beeinflusst die Dimensionierung nach Standardregeln zwar QoS-Metriken, jedoch QoE nur marginal.

Auf der Anwendungsebenen kompensieren zusätzliche Puffer Leistungsschwankungen, die z.B. aus Netzwerkpuffern resultieren. Ein weit verbreiteter IPTV-Dienst nutzt solche Puffer, um Verluste mittels einem proprietären Retransmissionprotokoll zu kompensieren und die Videoqualität zu steigern. Eine Studie gibt Einblicke in die Funktionsweise dieses Protokolls und motiviert dadurch die Erweiterung von QoE Metriken, die solche Korrekturmaßnahmen üblicherweise nicht vorsehen und daher zu Fehlabschätzungen führen können. Zur Optimierung von Web QoE untersucht eine weitere Studie die Trefferraten von Cachingverfahren unter Berücksichtigung von YouTube Abfragehäufigkeiten. Dabei wird ein optimiertes Verfahren diskutiert, dass höhere Trefferraten ermöglichen kann, als bei herkömmliche LRU-Caches.

Abschließend betrachtet die Dissertation E-Mail Spam als einen relevanten QoE Einflussfaktor. In dieser Studie wird Address Harvesting als Ursache von Spam und verschiedene Mechanismen zur Spamvermeidung, mit dem Ziel der E-Mail QoE Optimierung, über einen Zeitraum von 3,5 Jahren untersucht.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Contributions | 3 |
| 1.2 | Thesis Structure | 6 |
| 2 | Background | 9 |
| 2.1 | Internet Architecture | 9 |
| 2.2 | Internet Traffic | 11 |
| 2.2.1 | Protocol & Application Mix | 11 |
| 2.2.2 | Statistical Traffic Properties | 12 |
| 2.2.3 | Traffic Generation | 12 |
| 2.3 | Internet Measurement | 13 |
| 2.3.1 | Basic Principles | 13 |
| 2.3.2 | Practical Challenges | 14 |
| 2.3.3 | Measurement Tools | 15 |
| 2.4 | Network Buffering in the Internet | 15 |
| 2.4.1 | Packet Buffers | 15 |
| 2.4.2 | Packet Buffer Sizing | 16 |
| 2.4.3 | Bufferbloat | 17 |
| 2.4.4 | Packet Buffer Locations | 18 |
| 2.5 | Application Layer Buffering | 19 |
| 2.6 | Quality of Service | 20 |
| 3 | Quality of Experience | 23 |
| 3.1 | Definition | 23 |
| 3.2 | QoE Model Construction | 25 |
| 3.3 | QoE Model Application | 26 |
| 3.3.1 | Speech QoE | 27 |
| 3.3.2 | Video QoE for UDP/RTP Streaming | 28 |
| 3.3.3 | Video QoE for TCP/HTTP Streaming | 31 |
| 3.3.4 | Web QoE | 31 |
| 3.4 | Long-Term QoE Integration | 32 |
| 3.5 | Discussion | 33 |

| | | |
|----------|---|-----------|
| 4 | QoE Management of UDP Video Streaming | 35 |
| 4.1 | Packet Loss Impact on UDP Video Streaming | 35 |
| 4.1.1 | Measurement Methodology | 36 |
| 4.1.2 | Packet Loss Influence | 37 |
| 4.1.3 | Trade-off between Packet Loss and Content Quality | 38 |
| 4.1.4 | Discussion | 39 |
| 4.2 | Bandwidth Reduction by Scalable Video Encoding | 40 |
| 4.2.1 | Scalable Video Encoding | 41 |
| 4.2.2 | Measurement Setup | 41 |
| 4.2.3 | Video Resolution | 42 |
| 4.2.4 | Frame Rates | 43 |
| 4.2.5 | Trade-Off between Bandwidth, Frame Rate and Resolution Reduction | 44 |
| 4.3 | QoE Management | 44 |
| 4.4 | Related Work | 45 |
| 4.4.1 | Prior Work | 45 |
| 4.4.2 | Follow-up Work | 46 |
| 4.5 | Discussion | 47 |
| 4.6 | Future work | 47 |
| I | Impact of Network Buffer | 51 |
| 5 | QoE Impact of Packet Loss Models | 53 |
| 5.1 | Markovian Loss Models | 54 |
| 5.2 | Second-order Statistics of Loss Processes | 55 |
| 5.2.1 | Empirical DVB-H Loss Trace | 57 |
| 5.2.2 | Parameter Estimation | 57 |
| 5.3 | Video Quality Evaluation of Impaired Video Sequences | 58 |
| 5.3.1 | Freezing | 59 |
| 5.3.2 | Slicing | 59 |
| 5.3.3 | Influence of Fitting Method | 61 |
| 5.4 | Related Work | 62 |
| 5.5 | Discussion | 63 |
| 5.6 | Future Work | 64 |
| 6 | QoE Impact of Packet Buffers | 65 |
| 6.1 | Buffering in the Wild | 66 |
| 6.2 | Measurement Setup | 69 |
| 6.2.1 | Testbed Setup | 69 |
| 6.2.2 | Traffic Scenarios | 74 |
| 6.2.3 | Buffer Configurations | 75 |
| 6.3 | QoS: Background Traffic | 76 |

| | | |
|-----------|---|------------|
| 6.4 | Voice over IP | 78 |
| 6.4.1 | Approach | 78 |
| 6.4.2 | Access Networks | 79 |
| 6.4.3 | Backbone Networks | 82 |
| 6.4.4 | Key Findings for VoIP QoE | 83 |
| 6.5 | RTP Video Streaming | 83 |
| 6.5.1 | Approach | 84 |
| 6.5.2 | Access Networks | 85 |
| 6.5.3 | Backbone Networks | 86 |
| 6.5.4 | Key Findings for RTP Video QoE | 87 |
| 6.6 | Web Browsing | 87 |
| 6.6.1 | Approach | 88 |
| 6.6.2 | Access Networks | 88 |
| 6.6.3 | Backbone Networks | 90 |
| 6.6.4 | Key Findings for WebQoE | 92 |
| 6.7 | Excursion: YouTube / HTTP Video Streaming | 92 |
| 6.7.1 | Approach | 92 |
| 6.7.2 | Backbone Networks | 93 |
| 6.8 | Related Work | 95 |
| 6.9 | Discussion | 96 |
| 6.10 | Future Work | 97 |
| II | Impact of Application Buffer | 99 |
| 7 | QoE Impact of Retransmission Buffers in IPTV Set-Top-Boxes | 101 |
| 7.1 | Measurement Setup | 102 |
| 7.2 | Simulation Design and Buffer Models | 104 |
| 7.2.1 | De-Jitter/Playout Buffer Model | 104 |
| 7.2.2 | Implemented ARQ Schemes | 105 |
| 7.3 | Evaluation of Set-Top Box Behavior | 106 |
| 7.3.1 | Request Packets | 106 |
| 7.3.2 | Retransmission Packets | 108 |
| 7.3.3 | Comparing Correction Efficiency | 108 |
| 7.3.4 | Comparing Induced Network and Server Load | 110 |
| 7.3.5 | QoE Impact | 110 |
| 7.4 | Related Work | 112 |
| 7.5 | Discussion | 112 |
| 7.6 | Future Work | 113 |
| 8 | HTTP Caching | 115 |
| 8.1 | Cache Placement Scenarios | 117 |
| 8.1.1 | Caching Benefit | 117 |
| 8.1.2 | Broadband Access Networks | 118 |

| | | |
|-----------|--|------------|
| 8.1.3 | Mobile Access Networks | 119 |
| 8.2 | Caching Schemes | 120 |
| 8.2.1 | Notation | 120 |
| 8.2.2 | Most Popular Objects | 121 |
| 8.2.3 | Least Recently Used | 122 |
| 8.2.4 | Sliding Window | 122 |
| 8.3 | Evaluation of Cache Hit Rates | 123 |
| 8.3.1 | Data Set | 123 |
| 8.3.2 | Simulation Methodology | 124 |
| 8.3.3 | Least Recently Used | 125 |
| 8.3.4 | Sliding Window | 126 |
| 8.3.5 | Traffic Reductions | 128 |
| 8.4 | Related Work | 131 |
| 8.5 | Discussion | 132 |
| 8.6 | Future Work | 133 |
| 9 | Excursion: E-Mail Spam & Address Harvesting in the Internet | 135 |
| 9.1 | Methodology & Datasets | 136 |
| 9.2 | Harvest and Spam Activities | 138 |
| 9.2.1 | Network Level Properties | 138 |
| 9.2.2 | E-Mail Address Usage | 139 |
| 9.3 | Spam Mitigation | 140 |
| 9.3.1 | Fingerprinting: User Agent Strings | 140 |
| 9.3.2 | Address Presentation Method Robustness | 142 |
| 9.3.3 | Efficiency of Blacklisting & Usage of Anonymity Services | 142 |
| 9.3.4 | Harvesting: Role of Search Engines | 143 |
| 9.4 | Related Work | 143 |
| 9.5 | Discussion | 144 |
| 9.6 | Future Work | 145 |
| 10 | Conclusion and Outlook | 147 |
| 10.1 | Summary | 147 |
| 10.2 | Future Directions | 149 |
| | List of Figures | 153 |
| | List of Tables | 155 |
| | Bibliography | 157 |

Published Papers

Parts of this thesis are based on the following peer-reviewed papers that have already been published. All my collaborators are among my co-authors.

International Conferences

HOHLFELD, O., GRAF, T., AND CIUCU, F. Longtime Behavior of Harvesting Spam Bots. In *ACM Internet Measurement Conference* (2012).

HOHLFELD, O., BALARAJAH, B., BENNER, S., RAAKE, A., AND CIUCU, F. On revealing the ARQ mechanism of MSTV. In *IEEE International Conference on Communications (ICC)* (2011)

HASSLINGER, G., AND HOHLFELD, O. Efficiency of caches for content distribution on the Internet. In *International Teletraffic Congress* (2010).

International Workshops

ZINNER, T., ABOUD, O., HOHLFELD, O., AND HOSSFELD, T. Impact of frame rate and resolution on objective QoE metrics. In *IEEE Workshop on Quality of Multimedia Experience* (2010).

ZINNER, T., ABOUD, O., HOHLFELD, O., HOSSFELD, T., AND TRAN-GIA, P. Towards QoE management for scalable video streaming. In *ITC Specialist Seminar on Multimedia Applications - Traffic, Performance and QoE* (2010).

HOHLFELD, O., AND CIUCU, F. Viewing impaired video transmissions from a modeling perspective. *ACM SIGMETRICS MAMA. Performance Evaluation Review* 37, 2 (Sept. 2009), 33–35.

Technical Reports

HOHLFELD, O., PUJOL, E., CIUCU, F., FELDMANN, A., AND BARFORD, P. BufferBloat: How Relevant? A QoE Perspective on Buffer Sizing. *Technical Report 2012-11*, TU Berlin, 2012.

Under Submission

Parts of this thesis are based on the following paper that is currently under submission.

International Conferences

HOHLFELD, O., PUJOL, E., CIUCU, F., FELDMANN, A., AND BARFORD, P. A QoE Perspective on Sizing Network Buffers. Submitted to *ACM IMC* (2013)

1

Introduction

The Internet, which is often perceived only as a single entity, actually is a network of networks whose structure challenges service level deployment. This meta network is composed of more than 42,000 individual networks (Autonomous Systems) [132], each forming separate administrative domains. An immediate consequence of this interconnected nature is the need for business-level agreements to organize separately managed networks into one meta network. On the one hand, this aspect manifests in the economics of network interconnections that are either settlement free (e.g., peering links) or attached to transit fees (e.g., customer-provider links). On the other hand, the current structure and design of the Internet hinders the deployment of Internet-wide service level guarantees. This challenge is rooted in technical and business level aspects [61]. From a technical perspective, Internet-wide Quality of Service (QoS) support is hindered by *i*) the state complexity of resource reservations and *ii*) the lack of QoS support in inter-domain routing. From a business perspective, enabling QoS requires bilateral agreements amongst all participating networks. As a result, QoS mechanisms are typically used *within* autonomous systems (intra AS) to deploy service levels for internal services (e.g., IPTV, VoIP, or BGP sessions), but not *between* networks (inter AS). Thus, despite the widespread use of the Internet, enforcing service levels that help to enable high user satisfaction throughout the Internet still is a challenging problem.

This lack of Internet-wide QoS particularly challenges the deployment of real-time and multimedia services. Multimedia services are increasingly popular and already account for significant traffic volumes [176, 96, 19]. In particular, Cisco [19] forecasted video to account for 55% of the overall consumer Internet traffic in 2016. However, in contrast to elastic traffic (e.g., downloads), real-time and multimedia

services have more stringent service level requirements to function properly. Failures to meet the posed requirements can result in user dissatisfaction with the offered service (see e.g., [20]).

Accounting for user satisfaction is important as Internet users expect good service quality [194]: services should be always reachable and react fast, video should be streamed without visual artifacts, and voice calls should not be impacted by audio impairments. The user perception and satisfaction can be assessed by Quality of Experience (QoE) metrics. As QoE depends both on the users' perception and the used service, they form end-to-end metrics. While network optimization has traditionally focused on optimizing network properties such as QoS, we focus on optimizing end-to-end QoE metrics in this thesis. We argue that end-user QoE is the measure that is relevant for network operators and service providers.

QoE depends on a multitude of aspects including physical network properties and link speeds, transport protocols, or processing capabilities in network elements and servers. In addition, within a network, QoS mechanisms are often necessary to improve QoE by bounding QoS metrics such as delay, jitter, and packet losses. As such, the absence of QoS mechanisms—e.g., for applications that are deployed Internet wide (over the top services)—challenges the delivery of high QoE. As a consequence, optimizing QoE is a challenging cross-layer optimization problem that is not yet fully understood.

Interactive applications such as VoIP or gaming have stringent requirements regarding low delays. In addition, several studies suggest that the importance of low delays extends to other applications including browsing [175, 226, 52, 237, 170, 42]. An increase in latency reduces user interactions with web sites and thus reduces revenue. For example, an increase in the processing delay of 100 ms (400 ms) reduced the daily amount of Google searches per user by 0.2% (0.4%), respectively [226, 52]. Schurman reports similar findings for Microsoft Bing and reports a revenue decline of 0.6% (2.8%) for 500 ms (1000 ms) of additional latency [226]. Sefanov reports a 5% to 9% traffic drop for a latency increase of 400 ms at Yahoo [237]. Linden found that a 100 ms increase in latency drops the sales rate at Amazon by 1% [170]. These examples highlight that even small delay increases can have a significant business impact.

The need for speed also fostered new business models such as Content Delivery Networks (CDN). CDNs exploit the existence of content whose popularity is Zipf distributed and operate a geographically distributed network of caches to bring content closer to end-users—and thus serve it with low latency. Also Internet Service Providers have tried to monetize delay reducing options (e.g., fast-path DSL) that promise better QoE. Despite this effort, recent observations demonstrated the possibility of delays in the order of seconds caused by excessive queueing delays (i.e., bufferbloat) [16, 158, 70].

Buffering occurs in *packet buffers* located in core and edge routers and switches, but also in software at the *application layer*. Packet buffers help to increase throughput by absorbing transient traffic bursts and thus reducing negative effects of packet loss. However, large buffers can induce significant delays and thus degrade the performance of network applications; small buffers, in turn, can have inverse effects. Surprisingly, even after decades of research and operational experience, ‘proper’ buffer dimensioning remains challenging due to trade-offs in network performance metrics. Recently, the debate has focused on the claim that excessive buffering (*i.e.*, *bufferbloat*) can “lead to network-crippling latency spikes” [3] which severely impact Internet services. Excessive buffering in the edge has been reported to contribute delays in the order of seconds [16, 158, 70].

In addition, applications deploy *application layer* buffering to address latency problems mainly originating from packet buffers at the network layer, e.g., jitter. De-jitter buffers in multimedia applications absorb delay variations (jitter) that impose challenges for multimedia applications, which need constant playout times for audio and video frames.

Further, caching reduces the retrieval delay of frequently accessed content.

1.1 Contributions

Thesis Statement. The application of Quality of Experience concepts provides a new perspective on buffering that differs from the classical perspective of using network performance metrics such as jitter, delay, packet loss, throughput, link utilization, etc. This is particularly relevant for router packet buffers in which only narrow conditions seriously impact QoE.

We next summarize the main main focus areas of this thesis.

Quality of Experience Assessment. Quality of Experience is an active research area whose core objective is to quantify the users’ perception of the applications or of services. This is challenging since the users’ perception is subjective. Approximative QoE models allow a mapping of influence factors (e.g., packet loss) to subjective quality scores by using mappings established in subjective tests. This thesis reviews common QoE models from the perspective of practitioners in network research.

The reviewed concept are systematically applied to evaluate buffer influences. A first evaluation quantifies influences of video resolution, frame rate, scaling method, and video content on objective quality metrics. An illustration highlights that packet loss severely degrades objective QoE scores and should be avoided. Motivated by this observation, this thesis discusses bandwidth reduction mechanisms to avoid packet losses in congested networks (e.g., resulting from overflowing buffers). Bandwidth reductions can be achieved by either *i*) video resolution reductions, *ii*) image quality reductions in terms of coding and bitrate, or *iii*) frame rate reductions. In this

context, this thesis evaluates Scalable Video Coding that is able to seamlessly adapt to varying network conditions. This evaluation particularly focused on investigating impacts of partial scalability on objective quality metrics.

QoE Impact of Network Buffers. To model the packet loss process resulting from overflowing packet buffers, this thesis studies probabilistic packet loss models. Such model-based loss generators are used in QoE studies to produce deterministic error patterns. We evaluate the QoE-impact of different models and show that model choice matters. This leads us to evaluate a new fitting technique that is optimized for replicating aspects relevant to video QoE.

The size of network buffers influences network performance by controlling the level of induced delay, jitter, and packet loss. The question of ‘proper’ buffer dimensioning remains an unresolved and controversially discussed aspect. Most recently, this discussion has focused on the negative effects of large buffers. This led to proposed Internet engineering changes, despite the absence of sufficient empirical evidence. As an understanding of buffering effects is crucial before altering important engineering aspects, this thesis broadly studies the impact of buffer sizing on QoE metrics in an extensive study involving relevant user applications (e.g., voice, video, and web browsing), real hardware, and realistic workload. This study shows that the dominant factor for the QoE is the level of competing network workload. That is, a workload configuration for which the competing flows keep the queue at the bottleneck link filled (e.g., via many short-lived and therefore not congestion controlled flows) have much larger impact on QoE than buffer size. The study additionally shows that exacerbated (bloated) buffers have a significant effect on QoE metrics. Reasonable buffer sizes that follow standard sizing guidelines, however, have a significant effect on QoS metrics, but impact QoE metrics only marginally. This leads us to conclude that limiting congestion, e.g., via QoS mechanisms or over-provisioning, may actually yield more immediate improvements in QoE than efforts to reduce buffering.

QoE Impact of Application Buffers. Application buffering is used to compensate for performance variations, e.g., originating from network buffering. In this context, we focus on two scenarios. In the first scenario, we study the proprietary retransmission scheme deployed in a major IPTV network. We show that the resend mechanism deployed by Set-Top-Boxes in a major IPTV network is based on a simple buffer-based resend scheme that offers drastic QoE improvements for low loss rates. When QoE metrics are used by ISPs for QoE monitoring inside the network, they do not account for error recovery mechanisms at the edge and thus are prone to QoE mispredictions. By revealing insights into the resend mechanism used by a major IPTV network, we pave the way for improved QoE metrics accounting for error recovery mechanisms.¹

¹An IPTV monitoring tool released after our study monitors the resend traffic to improve QoE predictions.

In the second scenario, we discuss HTTP caching to optimize Web QoE. In this study, we performed a hit rate analysis of caching schemes by focusing on YouTube video popularities. HTTP video delivery, e.g., by YouTube, accounts for significant traffic volumes [176] and shows promising cachability [101]. Caching offers potential for increasing quality by reducing latencies and link loads, which in turn help avoiding congestions that we show to have detrimental quality impacts. While the concept of caching differs from buffering, they represent related problems, e.g., as the size of a cache impacts its efficiency. This leads us to evaluate caches of a broad range of sizes. We further contributed a new caching scheme that offers higher cache hit rates than traditional Least Recently Used caches.

Spam Mitigation. With an estimate of 2,4 billion users worldwide [243], e-mail is the most widespread communication service in the Internet. In this service, the presence of unsolicited bulk e-mail (spam), which has exceeded the volume of legitimate e-mail, remains a costly economic problem that largely degrades the QoE of e-mail communication. In a further analysis, we broaden our view by looking at a major communication system and spam as one of its main QoE determinants. By this, we explore an area that has not yet been considered in QoE research. A large-scale study conducted over 3.5 years reveals insights into address harvesting as the origin of spam and proposes mechanisms for spam mitigation. The study shows that harvester can often be identified by their HTTP user agent string and that simple address obfuscation methods sufficiently protect e-mail addresses from being harvested. In addition to classical spam filtering efforts, this study highlights that simple spam prevention methods can help to improve the QoE of e-mail.

Summary of Contributions

To summarize, the highlights of this thesis are as follows.

- It broadly investigates the influence of buffering in the network as well as at the application layer on Quality of Experience.
- It presents the first comprehensive study on the impact of buffer sizes on *Quality of Experience*. Concretely, it shows that the dominant factor for the QoE is the *level of competing network workload*. That is, workloads in which the competing flows keep the queue of the bottleneck link filled have a much larger impact on QoE than buffer size. In addition, it shows that exacerbated (bloated) buffers have a significant effect on QoE metrics. Reasonable buffer sizes that follow standard sizing guidelines have a significant effect on QoS metrics, but impact QoE metrics only marginally. This lead us to conclude that limiting congestion, e.g., via QoS mechanisms or over-provisioning, may actually yield more immediate improvements in QoE than efforts to reduce buffering. In this way, this thesis paves the way for designing load dependent dynamic buffer sizing schemes.

- Regarding the application layer, this thesis makes two contributions. We first show that a simple resend mechanism is deployed by Set-Top-Boxes in a major IPTV network that offers QoE improvements. To optimize Web QoE, we performed a hit rate hit rate analysis of caching schemes by focusing on YouTube video popularities. The thesis contributes an optimized caching scheme that offers higher cache hit rates than traditional Least Recently Used caches.
- An extension of the scope towards e-mail as the most widespread communication service discusses Spam as a major QoE determinant. By this discussion, we broaden the view of QoE that currently does not study e-mail QoE.

1.2 Thesis Structure

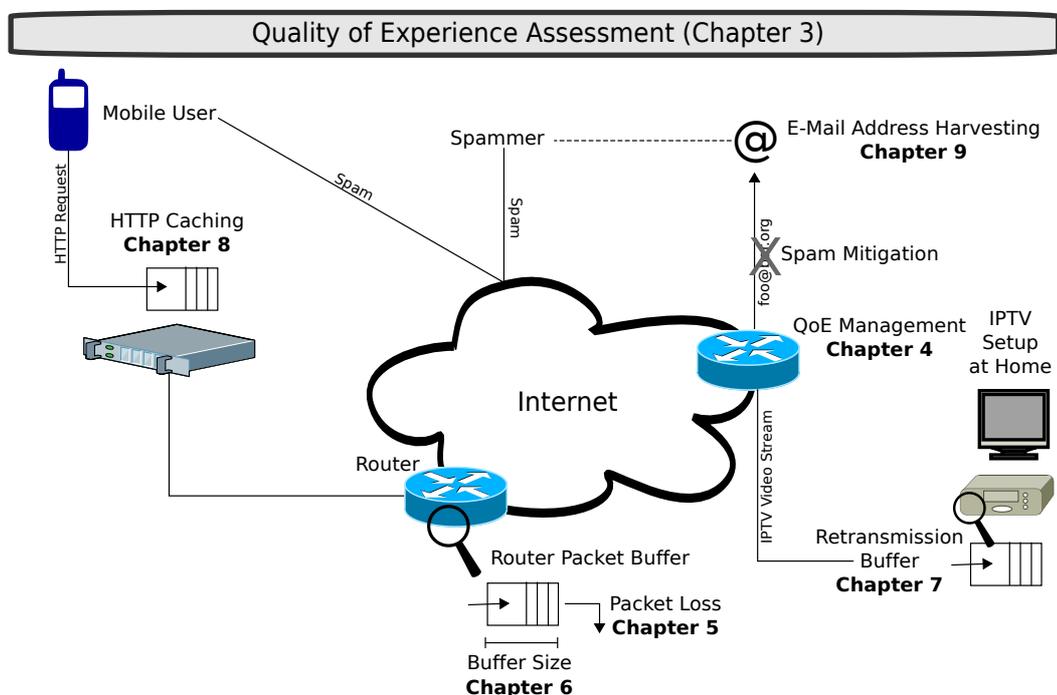


Figure 1.1: Thesis overview

The thesis structure is illustrated in Figure 1.1. In Chapter 2, we begin by briefly reviewing basic concepts used in the course of this thesis.

We then focus on the user perception of application quality in Chapter 3. This chapter first reviews Quality of Experience (QoE) estimators that provide automatic assessments for aspects of user perception. Chapter 4 then uses the introduced QoE assessment algorithms to evaluate factors impacting video streaming QoE. This

evaluation leads us to discuss techniques for optimizing QoE in video streaming services.

In the first part of this thesis, covering Chapters 5 to 6, we discuss router buffers on the Link- / Network Layer. Chapter 5 focuses on the packet loss process as outcome of packet buffers. It introduces a probabilistic packet loss model that is optimized for QoE. Chapter 6 discusses the influence of packet buffer size on QoE in general.

We then move on to discuss the influence of application layer buffering in the second part of the thesis. Chapter 7 reveals insights in the retransmission mechanisms of Set-Top-Boxes in a major IPTV network and their QoE impact. Chapter 8 discusses HTTP caching in light of YouTube video transmissions.

Chapter 9 broadens the view of this thesis by presenting an excursion to e-mail as the most widespread communication service in the Internet and spam as arguably a major QoE determinant.

Chapter 10 concludes the thesis and summarizes our results. Based on our findings, we discuss directions for future research.

2

Background

The Internet¹ as a network of networks has evolved to a complex structure. This chapter reviews this structure and its evolution. We highlight relevant properties of the Internet, traffic properties, and means for Internet measurement. After reviewing the fundamental concepts, we discuss the role of buffering in the Internet. We then discuss how the complex structure of the Internet currently challenges the deployment of Internet wide Quality of Service (QoS) mechanisms that help to meet application demands. This is relevant as dismissing application demands can result in low application performance and in bad user experience (Quality of Experience).

2.1 Internet Architecture

The Internet; what typically is perceived only as a single entity, actually is composed of more than 42,000 individual networks (Autonomous Systems) [132] that are interconnected in more than 300 Internet eXchange Points (IXP) [83] and other peering facilities. To function as one joint network, the individual subnetworks are glued together by the Border Gateway Protocol (BGP). We next discuss structural properties of this network of networks and how they evolved over time.

The traditional AS-level model of the Internet defines a hierarchical, tiered structure of networks with a relatively small number of interconnected tier-1 networks at its

¹We focus on the Internet as the dominant packet-switched network. However, many of the discussed techniques and properties apply to packet-switched networks in general (e.g., traffic self-similarity, measuring techniques, traffic generation, buffering, and Quality of Service.).

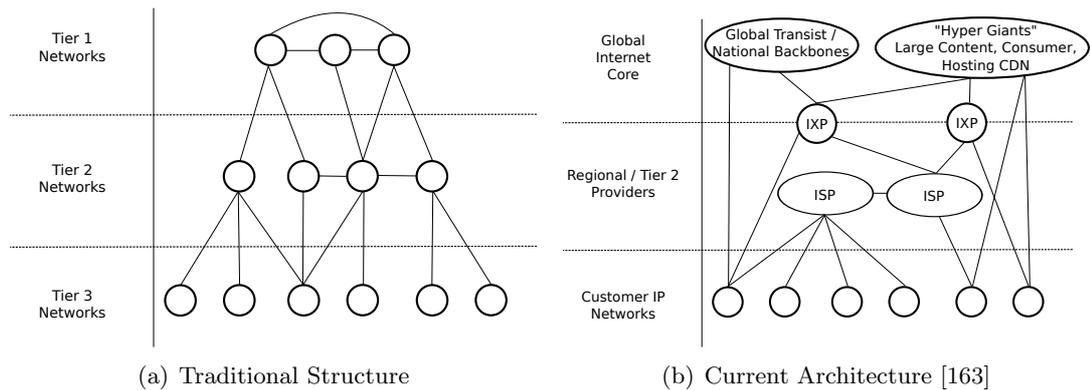


Figure 2.1: Logical structure of the Internet

top [162] (see Figure 2.1(a)). Tier-2 networks provide regional or national coverage and are connected to only a few tier-1 networks. The bottom contains lower-tier networks that provide connectivity to other low-tier networks or to end users.

This mental model, however, has changed over time and resulted in a flattened topology. In the traditional structure of the Internet, content is served by lower-tier networks to end-users located in other lower-tier networks. Recent studies, however, report a consolidation of inter-domain traffic from well-connected content providers (e.g., Google, Akamai) that does not traverse the Internet’s core [163]. As a recent change, the advent of “hyper giants” (e.g., large content distribution networks) has flattened the hierarchy. Having large and well-connected “hyper giant” networks bypassing the traditional tiered structure flattens the topology and creates a new mental model of the Internet as outlined in Figure 2.1(b). While it was often believed that networks mostly peer in private peering facilities, recent work [29] highlighted the importance of IXPs as highly connected peering points.

An important aspect of the Internet’s structure is the interconnection of individual networks as individual administrative domains. This manifests in the importance of business-level agreements for the Internet to function as a large, interconnected meta network. The peering ecosystem denotes one example of business-level agreements in which monetary properties are attached to traffic traversing other networks / autonomous systems (ASes). For example, while peering links among tier-1 networks are settlement free, tier-2 networks can be customers of tier-1 networks and pay transit fees for providing Internet access (customer-provider relationship). The resulting diversity in business interests and the need for business-level agreements has further consequences for the Internet. In particular, the absence of a Quality of Service enabled Internet has major implications on end-user experience (see Section 2.6).

2.2 Internet Traffic

Since the establishment of the early ARPAnet in the 1970s, the size of the Internet [133, 132] as well as the carried traffic [241] is constantly growing. Within 12 years, the estimated number of Internet users has grown from 360 million users in 2000 to 2.4 billion users in 2012 [107]. Another example of this growth is the increase of the Google web page index that grew by a factor of 10,000 from 30 million indexed web pages in 1999 to billions of web pages nowadays [176]. The number of search queries issued to Google increased by a factor of 10,000 since 1999 [176] as well. This increase manifests a steadily increasing complexity resulting in higher demands for data processing and network capacity.

We next turn our attention to the nature of the traffic traversing the Internet. This section first discusses the protocol and application mix, followed by self-similarity as an important statistical property resulting from this mix. Afterwards, this section discusses how this knowledge can be used to generate realistic traffic, as employed later in the thesis.

2.2.1 Protocol & Application Mix

Driven by the increasing Internet penetration and the increasing bandwidth usage of modern applications, Internet traffic volume has been reported to grow exponentially at rates of 35% to 45% [163, 96, 80] per year. While traffic growth has been a rather stable pattern, the actual traffic composition is subject to trends and regional preferences.

The variety of used protocols and their respective usage differs from layer to layer. We exclude link layer protocols as they are not globally routed. The network layer defines globally routable Internet addresses and consists purely of the Internet Protocol (IP) used in version 4 and 6. The transport layer provides end-to-end communication services and consists of connectionless UDP and connection-oriented TCP as the major transport layer protocols. On the transport layer, the dominant portion of the Internet's traffic is attributed to TCP [23, 172], which accounts for $\approx 80\%$ [23] of the traffic volume.

While the transport layer mainly consists of the UDP and TCP protocols, the application layer exhibits a much wider variety of protocols used in the wild. This results from the End-to-End Argument [162] as founding principle of the Internet. The argument states that application-related mechanisms should reside in the network edge rather than in the network core, e.g., as opposed to the telephone network. This design principle is a major driver of innovation, as it allows the creation of new applications without explicit support by the network core. As a consequence of the resulting innovation process, application protocol usage is largely subject to trends and regional preferences.

While new applications and protocols come and go, two major traffic shifts were reported in the last two decades. After the advent of the HTTP protocol in 1992, it accounted for the dominant portion of the Internet’s traffic [65]. The dominance of HTTP declined in the late 1990s with the advent of peer-to-peer (P2P) systems. For several years, many studies accounted significant traffic shares to P2P traffic [96, 48, 202, 224, 225]. For instance, the 2007 Ipoque study reported P2P traffic to account for 49% to 83% of the overall Internet traffic [224]. The dominance of P2P traffic declined in the mid 2000s and HTTP became again the dominant application layer protocol [29, 48, 96, 172, 163, 23, 80].

2.2.2 Statistical Traffic Properties

The advent of Erlang’s model of the Copenhagen telephone exchange marked the beginning of what is now known as queueing theory. Back then, call arrivals were assumed to be exponentially distributed as they were independently generated by a large population of users [47]. Also call durations were assumed to follow an exponential distribution.

In the early days of the Internet, traffic in packet networks was believed to exhibit a similar pattern as calls in the circuit-switched telephony network [200]: packet arrivals and their sizes were commonly assumed to be exponentially distributed. This picture changed drastically when aggregated traffic was found to exhibit self-similar patterns [167]. Similar to fractal geometry, where the appearance of fractal objects remains constant—or self-similar—over scale, traffic bursts were found to have no natural length and can be observed over multiple time-scales. Unlike the previously believed Poisson traffic, aggregating packet traffic does not smooth-out its burstiness. This finding largely invalidated previously used modelling assumptions [200] and created the need to revisit theoretical assumptions and models of the Internet.

Self-similar traffic is created by aggregating ON/OFF sources whose ON and OFF period lengths are heavy-tailed [261]. One way to achieve this property is by transferring files and web sites whose sizes follow heavy-tailed distributions [66].

2.2.3 Traffic Generation

Based on previously discussed empirical findings, we next discuss properties of realistic Internet-like traffic generation. In this thesis, we mainly generate traffic according to properties observed in measurement traces in simulations or testbeds. Other evaluations (e.g., as in Chapter 7) are based on impairing existing traffic and thus do not need traffic generation.

In testbed-driven evaluations (see e.g., Chapter 6), we generate synthetic traffic based on pre-defined properties. For this traffic generation, we use the Harpoon

flow level network traffic generator [234] to create realistic traffic, i.e., self-similar TCP traffic with Internet like properties. Harpoon reproduces flow-level properties of Internet traffic and thus allows us to generate self-similar traffic in a laboratory setting. One of these properties to generate realistic traffic is the use of the systems native network stack by Harpoon. Another property is the use of sessions to mimic the behaviour of a user. Sessions generate flows and are configured with a specific interarrival time distribution, a file size distribution, and other parameters. The amount of traffic can be controlled by specifying the number of parallel sessions.

In contrast to operating a real network with a real software stack, simulations focus on simulating selected features. As simulations are specific to each studied problem, we will introduce their details in the respective chapters. For instance, the simulation conducted in Chapter 8 is based on measured YouTube video popularities to simulate cache efficiency.

2.3 Internet Measurement

Internet Measurement denotes a set of techniques to measure properties of the Internet. Properties that concern this thesis are traffic properties and buffering statistics. This section will review basic principles of measurement studies and discuss practical challenges that arise when conducting such studies. We close this discussion by reviewing essential measurement tools used in this thesis.

2.3.1 Basic Principles

In measurement studies, traffic is being captured at one or many *vantage points*. It is important to operate vantage points at relevant places in the network that allow the desired traffic to be captured. For instance, it is pointless to study the usage of HTTP in Germany from a vantage point located in Asia. The installation of vantage points is challenging, as it requires access to the measured network.

Measurement studies vary by the approach taken to collect traffic:

Active measurements actively inject probe traffic to perform measurements. The active nature allows taking control over the generated traffic, which is useful when studying situations that do not naturally occur in networks (e.g., worst-case performance by penetration testing).

Passive measurements do not inject traffic but passively capture traffic. As passive measurements do not interfere with the measured network, they give important insights into the functioning and usage of the network (e.g., application mix of user traffic, or buffer utilization).

2.3.2 Practical Challenges

Conducting large-scale Internet Measurement studies involves tackling a set of practical challenges [65] as discussed below.

Data volume: Capturing full-packet traces at modern link speeds has become a non-trivial problem due to disk IO limitations of capturing hardware. For instance, capturing a fully utilized 10 Gbps backbone link requires storing up to 1.2 GB/sec of data, or up to 4.3 TB/hour. Upcoming 100 Gbps backbone links will increase the data volume even further by a factor of 10.

Fortunately, links are rarely run at full capacity over extended time periods, which limits the amount of required data processing. Nevertheless, dealing with large data-sets provides a challenging engineering problem that can be tackled by a set of workarounds and optimizations. For example, if disk space is a concern in longitudinal studies, analyses with low computational complexity and low memory footprint can be performed online by only storing results and no unprocessed data. If an online analysis is not possible and access to packet payload is not required, the data volume can be reduced by only writing packet headers to disk. If the data volume is still too large, the data needs to be sampled during the capturing process. Sampling may limit the scope of analyses that can be performed on the data.

Anonymization: Dealing with network traffic data often involves proper anonymization [193, 63, 86, 65] of sensitive information in a privacy preserving manner. While this might look like an easily solvable problem at first, proper anonymization is challenging for two reasons.

1. Properties that allow de-anonymization of captured data are not always obvious. For example, based on U.S. census data from 1990 and 2000, it has been shown that birth day, ZIP code, and gender allow individuals to be identified with high confidence [103].
2. Anonymization should not render analyses infeasible. Anonymizing network traffic data often involves IP address anonymization. If the applied anonymization scheme is not prefix preserving [86] (e.g., by using a cryptographic hash function where two adjacent IPs are mapped to significantly different hashes), important network-level information on origination and destination networks are irrecoverably lost.

Data Availability: Conducting measurement studies requires the researcher to have access to data that helps in answering the posed research questions. However, data volume and privacy aspects largely prevent sensitive traffic measurement data from being shared and widely used in academic studies. One example of such data are traces containing traffic generated by a representative portion of Internet users. Such data can be captured at vantage points run by Internet

Service Providers (ISPs) oder Internet eXchange Points (IXPs). While such data allows the study of interesting Internet properties at large (e.g., its traffic mix and interconnection), the sensitive nature of customer generated data renders access to such data sets to be a hard problem. This is particularly challenging as complete anonymization is often not feasible as it would render the data unusable.

2.3.3 Measurement Tools

Internet Measurement relies on tools that scale with extensive data volumes found in modern traffic traces. For host-level captures, we rely on libpcap as API to capture traffic from the kernel-level networking stack. For accurate link-level captures we use DAG capturing cards² to provide accurate measurements. In contrast to host-level measurements, DAG cards have a high time resolution and perform packet timestamping in hardware.

We use Bro [199], tcpdump [5], and tshark [6] to extract protocol-level information from the captured traces. Bro allows analyzers to be written in Bro script and scales with extensive data volumes that span several TB. We use custom Perl, Python, and Shell scripts to perform post-processing of Bro, tcpdump, or tshark logs. Such scripts can be designed for scale by running online to compute and aggregate data, and to prevent slow disk IO as much as possible by sharing data in memory, e.g., by the use of Unix pipes.

2.4 Network Buffering in the Internet

Network buffering occurs in various places in the network (hosts, switches, and routers) and in various places in the network and software stack (interface hardware, device drivers, kernel). Most of these buffers are relatively small and do not significantly influence network performance. However, packet buffers, in particular in routers, can be large and impact network performance. Their performance impact is studied in the course of this thesis.

2.4.1 Packet Buffers

Packet buffers are widely deployed in network devices such as hosts, routers, and switches, mainly to fulfill two functions:

- storing an incoming bitstream in tiny buffers attached to physical interfaces until the packet is completely received and can be reassembled,

²<http://www.emulex.com/products/network-visibility-products/dag-packet-capture-cards/features/>

| Scenario | Link Speed | Buffer Size | | Delay |
|-------------------|------------|-------------|--------|----------|
| | | Packets | Size | |
| Access (Uplink) | 1 Mbps | 8 | 12 kb | 96 ms |
| Access (Uplink) | 1 Mbps | 256 | 375 kb | 3,072 ms |
| Access (Downlink) | 16 Mbps | 8 | 12 kb | 6 ms |
| Access (Downlink) | 16 Mbps | 256 | 375 kb | 192 ms |
| Backbone | 155 Mbps | 8 | 12 kb | 0.6 ms |
| Backbone | 155 Mbps | 65536 | 94 mb | 5,073 ms |
| Backbone | 10 Gbps | 8 | 12 kb | 0.01 ms |
| Backbone | 10 Gbps | 65536 | 94 mb | 79 ms |

Table 2.1: Queuing delays resulting from different buffer sizes assuming 1500 bytes packets

- reducing packet losses caused by transient bursts exceeding the egress links' capacity.

We focus our attention on the latter as those buffers can be large and impact the performance of the Internet. Due to its impact on the throughput achieved by transport protocols (see buff sizing).

Packets are dropped once the buffer capacity is exceeded. We model this loss process in Chapter 5. Using active queueing mechanisms such as RED [91] or CoDel [187] can result in early drops before the queue capacity limit is reached.

2.4.2 Packet Buffer Sizing

Besides dropping packets, buffering also delays packets and thus adds queueing delay that depends on the configured buffer size. We provide an intuition on the dependence of queueing delay on link speed and buffer size in Table 2.1. The settings in the table are motivated by our experimental infrastructure and typical configurations. The table shows a decay in queueing delays with faster links for the same buffer size. As such, access links can face significant queueing delays even though they are equipped with relatively small buffers.

Queueing delays can occur in two situations: *i*) in cases of *fast-to-slow transitions* somewhere in the end-to-end path, when switching to a slower (i.e., not fast enough) link, and *ii*) in cases of *congested* links. In effect, TCP flows will utilize the egress buffer at the bottleneck link assuming the flow is not window-limited and has sufficient data to send. Under any of these conditions, the data traversing the network will be buffered at the entry point of the bottleneck link.

Several rules have been proposed for buffer sizing. The rule-of-thumb [136, 250, 145] for dimensioning network buffers relies on the bandwidth-delay-product (BDP) $RTT * C$ formula, where RTT is the round-trip-time and C is the (bottleneck) link

capacity. The reasoning is that, in the presence of *few* TCP flows, this ensures that the bottleneck link remains saturated even under packet loss. This is not necessary for links with many concurrent TCP flows (*e.g.*, backbone links). It was suggested in [250] and convincingly shown in [35, 45] that much smaller buffers suffice to achieve high link utilizations. The proposal is to reduce buffer sizes by a factor of \sqrt{n} as compared to the BDP, where n is the number of concurrent TCP flows [35]. Much smaller buffer sizes have been proposed, *e.g.*, drop-tail buffers with $\approx 20 - 50$ packets for core routers [76]. However, these come at the expense of reduced link utilization [45]. For an overview of existing buffer sizing schemes we refer the reader to [251].

Despite the outlined advances in buffer sizing research, the BDP as rule-of-thumb is still manifested in IETF standards [145]. The outstanding update to engineering standards was addressed at the 86th IETF meeting held in March 2013 and resulted in a preliminary, work in progress version of an update to RFC 3819 [85, 84].

The implications of buffer size choices on application performance are largely unknown from technical, operational, economic, and even perceptual perspectives. Despite interest in the research community with regards to buffer dimensioning schemes, the issue remains far from resolved. This unresolved problem motivates us to broadly characterize the impact of buffering on Quality of Experience in Chapter 6.

2.4.3 Bufferbloat

Most recently, the buffer sizing debate has focused on the negative effects of large buffers. The essential argument is that excessive buffering in devices commonly deployed in the Internet today (*aka bufferbloat*) leads to excessive queuing delays (*e.g.*, in the order of seconds), which negatively influences the performance from a users' perspective [16]. Indeed, bufferbloat *can* adversely effect TCP by increasing round trip times or even triggering unnecessary TCP timeouts. It can also adversely effect UDP by increasing RTTs or packet losses.

The *existence* of excessive buffering has recently been shown at the network edge (*e.g.*, home routers and modems) [70, 158, 174, 240], end-hosts [3], and 3G networks [112]. These studies find that excessive buffering in the access network exists and *can* cause excessive delays (*e.g.*, on the order of seconds). This has fueled the recent bufferbloat debate [16, 98] regarding detrimental effects on network performance. We remark these findings must be interpreted with a degree of caution. While the studies discussed above clearly show the *existence* of large buffers, they do not show that they are actually being *used* in practice. We discuss the occurrence of buffer bloat in the wild in Section 6.1.

2.4.4 Packet Buffer Locations

We next discuss typical locations the end-to-end path that deploy buffers. Discussing commonly used buffer sizes in these locations is challenging as they are device, vendor, and configuration dependent. Moreover, network operators are reluctant to share configuration details. Despite this challenge, we also provide a *rough* intuition of expected buffer sizes. Due to the discussed challenges, this overview is not intended to be exhaustive.

End hosts: Buffering in end-hosts occurs in various places: in the kernel itself, in the device driver, and in the actual hardware. Buffers in a Linux host are in the order of several hundred up to a few thousand packets, introducing delays in the order of several ms up to a few seconds [3].

Home router: Several studies report excessive buffering in the edge, mainly concerning home routers and modems [70, 158, 174, 240]. The reported delays are in the order of several seconds. Fast-to-slow transitions often occur in the edge, when data is transferred to a typically slower access link. As a result, large buffers in home routers and modems contribute large queueing delays for traffic directed to the Internet (uploads). They are of less importance for pure downloads when the uplink is not utilized.

Access network: Buffering in the access network mainly occurs in switches that are typically equipped with a small amount of buffering, depending on the concrete device and its configuration.

Backbone network: Buffering in backbone networks mainly occurs in routers. Little is known about practically deployed buffer sizes except for sizing recommendations discussed in the previous section.

To illustrate the complex interconnection of buffers, we next discuss router buffers as one example architecture.

Packet Buffers in Routers

Routers are designed to interconnect networks. As such, they are equipped with multiple interfaces connecting different networks, a switching fabric interconnecting the interfaces, and processing units, e.g., used for route computation [39]. In larger, e.g., backbone-grade, routers, interfaces are hosted by linecards serving different layer 2 access technologies (e.g., Ethernet). Operations performed by a router fall into either control plane operations (routing), or data plane operations (packet forwarding / switching). In addition, routers are equipped with several types of memory that is typically used for *i*) running an OS and software, e.g., for route computation and maintenance, *ii*) keeping the forwarding table, and *iii*) for buffering packets.

The number of packet buffers, their concrete architecture, and their capacity varies between different networking devices and vendors. We next provide a concrete example by outlining the basic buffering architecture of Cisco 12000 backbone-grade routers [10]. Figure 2.2 shows selected buffers in the architecture. Packets arriving at a physical interface are first stored in small buffers of $2 \times \text{MTU}$ size, reassembled, and moved into larger *to fabric buffers*. Depending on the destination linecard, they traverse the switching fabric and are moved into a packet buffer specific to the outgoing interface (transmission queue, denoted as TX Queue in Figure 2.2). The size of the transmission queue impacts network throughput and has been subject to controversial debate, as discussed in the next subsection.

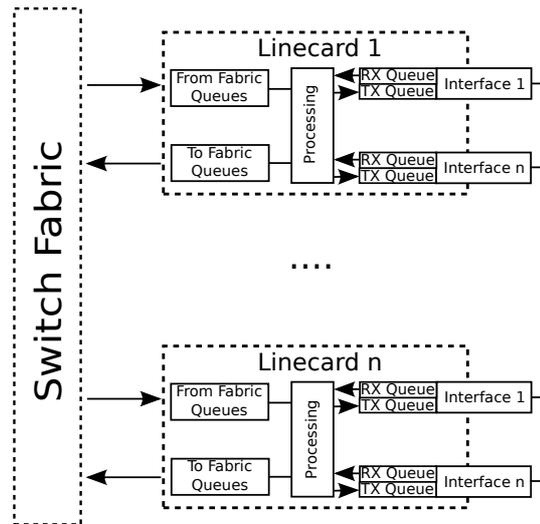


Figure 2.2: Buffer architecture of Ciscos 12000 backbone-grade router series

2.5 Application Layer Buffering

Buffering not only occurs in the network, but also in the applications. Application layer buffering mainly focuses on compensating varying network performance, e.g., loss and jitter caused by network buffering. This compensation aims at improving the application performance and thus the resulting QoE. We next discuss common types of application layer buffering.

De-jitter buffers are used in multimedia applications at receiver side to absorb jitter that results from processing delays and packet buffers. De-jitter buffers translate jitter into application layer packet loss when packets arrive after their scheduled playout time [137, 62, 239]. We do not investigate de-jitter buffers as their performance impact on multimedia applications has been extensively studied [213].

Retransmission buffers buffer data in multimedia applications to allow for retransmissions. Similar to de-jitter buffers, they delay the playout by the respective buffer depth to allow retransmitted packets to arrive before their scheduled playout time. A retransmission buffer can therefore be implemented using a de-jitter buffer. We discuss retransmission buffers in IPTV networks in Section 7.

Caches do not correspond to buffers as such, but represent a similar problem with similar characteristics. In contrast to buffering that holds data during transactions, caching stores data to accelerate subsequent requests. HTTP caching represents a popular application for reducing delay in HTTP transactions. We discuss the impact of HTTP caching on HTTP video transmission in Section 8. Similar to buffering, the size of a cache has implications for application performance.

2.6 Quality of Service

The Internet is designed as best effort network with no service level guarantees in mind. Quality of Service defines *additional* mechanisms aimed to provide service level guarantees that are particularly useful for real-time services to ensure user satisfaction. Quality of Service is defined by the ITU as the “totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs” [14]. Characteristics in this context denote observable and measurable parameters that are expressed by QoS metrics [14]. In IP networks, typical QoS metrics are delay, jitter, and packet loss³ [8]. Needs are application specific and are typically expressed by QoS metrics. For example, ITU-T Recommendation G.1010 [8] states that the one way delay for interactive voice communication should be below 150 ms and the packet loss rate should be below 3%. Other applications have less stringent QoS requirements, e.g., elastic-transfers such as bulk downloads via HTTP can tolerate higher delays and loss rates. On a contractual and business level, meeting pre-defined QoS levels is ensured by defining Service Level Agreements (SLAs) [173].

The absence of Internet-wide QoS support challenges QoS provisioning. This challenge is rooted in technical and business level aspects [61]. From a technical perspective, Internet-wide Quality of Service (QoS) support is challenged by *i*) the state complexity of resource reservations⁴ and *ii*) the lack of QoS support in inter-domain

³We remark that some amount of packet loss will always occur in IP networks running TCP. This is rooted in TCPs congestion control mechanism that uses packet loss for congestion signalling.

⁴Integrated Services allows the reservation of per-flow resources across a network. As it therefore needs intermediate routers to keep per-flow state, Integrated Services do not scale to an Internet-wide deployment.

routing⁵. From a business perspective, enabling QoS requires bilateral agreements amongst all participating networks. As a result, QoS mechanisms are currently used *within* autonomous systems (intra AS) to deploy service levels for internal services (e.g., IPTV, VoIP, or BGP sessions), but not *between* networks (inter AS). As long as no Internet-wide QoS is available, the Internet as best effort service will not always satisfy application demands.

Not meeting application demands can result in low application performance and in bad user experience (Quality of Experience). One example are complaints by Deutsche Telekom DSL customers due to high delays when playing online games [20]. The importance of service levels for maintaining customer loyalty and competitive advantages has been shown by Chiou [58]. This challenge motivates us to study application-specific Quality of Experience aspects and the relationship of QoS metrics and QoE in the next chapter.

⁵A multitude of solutions have been proposed in recent years to make inter-domain routing QoS-aware [264, 270, 44, 61]. However, none of the proposed techniques are in (widespread) use or left draft state [49].

3

Quality of Experience

The provisioning of broadband access for the mass market enables multimedia content to be extensively viewed and distributed over the Internet. Consequently, Quality of Experience (QoE) aspects, in terms of the service quality perceived by the user, become vital factors for ensuring customer satisfaction in today's networks. We argue that end-user QoE is the perspective that is relevant for network operators and service providers, and by extension, device manufacturers.

This chapter reviews the concept of QoE. We start by sketching how QoE estimators are constructed and then focus on reviewing a *selection* of the most relevant and often used QoE estimators for audio, video, and web browsing. These estimators form the basis for the QoE estimation which is the core of this thesis. We additionally discuss the relation between QoE and QoS metrics such as delay, jitter, and packet loss.

3.1 Definition

Quality of Experience (QoE) is “the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the users personality and current state.” [214]. To quantify the users' perception of

⁰The system theoretic view on QoE is joint work with Florin Ciucu and has been partially published in [121].

the quality of (network) applications, *QoE metrics*¹ have been defined for applications such as VoIP, Video, Web, etc. These metrics express quality mappings that were established in tests involving human subjects (see next section). Here, subjective quality ratings are often expressed on a 5-point absolute category rating scale ranging from ‘bad’ to ‘excellent’ [7]. We remark that this mapping involves using different techniques, including functions and (machine learning) algorithms that *do not* express metrics in the mathematical sense. For the sake of simplicity, we adopt the widely used (and sometimes criticized) term *QoE* or *quality metric* to express the concept of either a *metric*, a *measure*, or a *prediction algorithm*.

The user perception is influenced by network conditions and application specific properties, and can thus be expressed as a function of these

$$\text{Perception} = \mathcal{P}(\text{Network}, \text{Application}, \dots) , \quad (3.1)$$

for *some* mapping \mathcal{P} . Here, ‘Network’ reflects widely used QoS metrics which capture network conditions (*e.g.*, packet loss, delay, and jitter), and which implicitly include transport protocols, bandwidth limitations, or buffer sizes. ‘Application’ captures application-level properties such as video encoding artifacts, error concealment strategies, video resolution, and application layer buffering. QoE further depends on the perception of a service, the personality of the user (personality traits, gender, age, ...), and the users’ current state (*e.g.*, mood and expectations such as “premium calls should perform better than free calls”), and the current context (*e.g.*, free or paid call). It can be expressed as a function

$$\text{QoE} = \mathcal{T}(\text{Context}, \text{State}, \text{Personality}, \text{Perception}, \dots) , \quad (3.2)$$

for *some* mapping \mathcal{T} . Thus, QoE corresponds to a multidimensional perceptual space whereby its features are not necessarily independent of each other. We remark that our above discussion of QoE influence factors is simplified. As our focus is not on providing new QoE metrics, we restrict our discussion to an introduction of the general idea behind QoE modelling as needed to understand the application of QoE models in this thesis. For an extensive discussion on factors influencing QoE, we refer to [22].

Figure 3.1 illustrates this concept by introducing a dual-system representation of QoE. In the first system, the input signal x (*i.e.*, represents an encoded signal, such as, for example, a speech signal in a VoIP application context) is subject to the perturbations imposed by the network and its applications. Here, the network under study is controlled by u , *e.g.*, the network-workload, desired loss rate, or the buffer size configuration. The output (perturbed) signal y from the first system is then subject to the user perception. This is the represented by the second system.

¹We remark that *QoE metrics* do not correspond to metrics in a strict mathematical sense. By using the term *metric*, we follow the meaning of a performance measure, as it is widely adopted by the QoE and QoS communities. This measure can be a function, a (machine-learning) algorithm, or any other mapping.

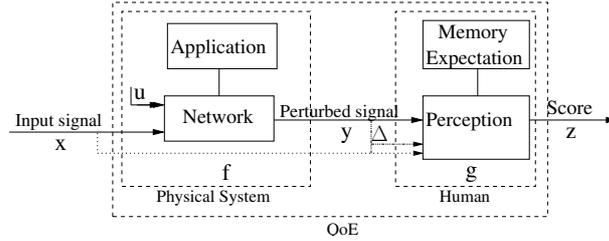


Figure 3.1: A dual-system view of QoE

Some equality estimates depend on differences between x and y , e.g., web page loading times, expressed as Δ . The output signal z then corresponds to the QoE judgment, e.g., expressed as a score on a 5-point scale. We again remark that Figure 3.1 presents a simplified set of QoE influence factors. For a more complete representation we refer to [22] for a list of influence factors and to [182, 213, 183] for an extensive discussion of audio-visual quality aspects.

We remark that QoE represents the ultimate perspective of the end-user. Thus QoE is not a system or network property, as for example the misleading usage of the term “network QoE” (see e.g., [38]) might suggest.

3.2 QoE Model Construction

The subjective nature of the *user perception* and quality formation process renders it hard to fully formalize \mathcal{T} . As such, a solid theoretical framework of QoE is still missing [22]. Current QoE models, metrics, and estimation algorithms can only be *approximative*.

Approximative QoE models / quality estimators are constructed as follows. Based on standard testing methodologies, an application, or signal created by an application, x is perturbed with n signals or system settings u_i 's, and the output (perturbed) signals y_i 's are further subject to the perception of k people (*i.e.*, the subjects) (see, e.g., ITU-T recommendations P.800 and P.910 [7, 15]), who provide quality ratings $z_{i,j}$:

$$(x, u_i) \rightarrow y_i \rightarrow (z_{i,1}, \dots, z_{i,k}) \quad \forall i = 1, \dots, n.$$

The individual scores $z_{i,j}$'s for subject j , source x , and setting i are then averaged in the so called Mean Opinion Score, $MOS_i = \sum_j z_{i,j}/k$. This reduction to the first moment can, however, lead to some inaccuracies in the resulting QoE models [127]. This mapping can be modelled by either using regression analysis or machine learning approaches.

Various relationships between u_i 's and z 's are considered in the QoE literature, including some that are consistent with results in psychology [220]. For instance,

| | Speech | Video | TCP | Web Browsing |
|-------------------|------------------------|---|---------------|-----------------|
| | | UDP | | |
| No Reference | E-Model [17] | Annoyance Prediction [87] Frame freezes [196] Loss visibility [142, 219, 143, 36] T-V-Model [215, 36] ITU P.1201.2 [21] | YouTube [128] | ITU G.1030 [11] |
| Reduced Reference | | Watermarking [77] | | |
| Full Reference | PESQ [9] POLQA [18] | PSNR [151, 186] SSIM [256] VQM [201] | | |

Table 3.1: Overview of QoE metrics

exponential relationships between the QoE score z and other parameters have been reported (IQX hypothesis²). Examples include data rates (x), packet loss rates (u_i) [220, 126], or web page loading times (Δ) [11]. In case of waiting times, the logarithmic dependency is in line with the Weber-Fechner law³. This, however, does not suggest that logarithmic relationships are a universal concept in QoE that always holds.

3.3 QoE Model Application

We next illustrate the *application* of QoE models to *approximate* the perceptive QoE score z . While the QoE literature provides a large body of approximative QoE models, we focus on reviewing a *selection* of what we consider to be the most relevant and often used QoE metrics for speech, video, and web browsing. We present an overview of relevant metrics in Table 3.1. For an extended overview of available metrics, we refer to [213, 183] for speech quality, to [37] for image quality, and to [77, 57, 64] for video quality.

QoE metrics can be classified into three categories by the required amount of reference information [262]. *Full-reference* (FR) metrics estimate the QoE score z based on the original signal x and received perturbed signal y . While FR metrics denote a desirable class of metrics, they are inapplicable when x is not available. This is often the case in network measurement. *No-reference* (NR) metrics account for this challenge and estimate z purely based on the received signal (e.g., based on Δ or y). As such, they rely on assumptions about x , e.g., the video content. Due to

²The IQX hypothesis states that the Interdependency between QoE and QoS follows an exponential function [129, 126, 90, 128].

³The differential perception dP is proportional to the relative differential $\frac{dS}{S}$ of a stimuli, whence the logarithmic perception-stimuli relationship is obtained by integration.

these assumptions, NR metrics have a tendency to be less accurate than FR metrics. *Reduced-reference* (RR) metrics account for this inaccuracy and estimate z based on a subset of features $f(x)$ and y .

We remark that we do not consider other situational factors such as the users' expectation (e.g., free vs. paid call) [183] which can also affect the perceived speech quality.

3.3.1 Speech QoE

Speech signals can be impaired by packet loss, jitter, and/or delay. To be more specific, packet losses directly degrade speech quality as long as forward error correction is not used. In addition to packet losses, network jitter can result in losses at the application layer as the data arrives after its scheduled playout time. This also degrades speech quality. Moreover, excessive delays impair any bidirectional conversation, as it alters the conversational dynamics in turn-taking behavior. We next discuss how these aspects can be measured.

Delay effects. Speech quality is impacted by two types of delay: *i*) pure delay impacting call interactivity and *ii*) talker echo. Delay alters conversational dynamics, e.g., for humans taking turns and/or interrupting each other. Turn taking is impaired by excessive delays and thus can significantly degrade the quality of the conversation [183, 150, 212, 222, 72, 113]. Thus, according to the ITU-T recommendation G.114, one-way delays should be below 150 ms (or at most 400 ms).

To capture the conversational quality of a phone call, the score z_d is computed as a function of the (measured) end-to-end delay Δ of the voice signal y , *i.e.*, (see [17]):

$$z_d = 100 - \begin{cases} 0 & \text{if } y \leq 100 \text{ ms} \\ 25 \left[\sqrt[6]{1 + \alpha^6} - 3\sqrt[6]{1 + \left(\frac{\alpha}{3}\right)^6} + 2 \right] & \text{otherwise ,} \end{cases}$$

where $\alpha = \frac{\log_{10} \Delta / 100}{\log_{10} 2}$. The range of this function is $[0, 100]$ (R scale), where 100 corresponds to excellent speech quality. This function describes the delay impairment factor of the E-Model [17]. We note that the notion of delay in the E-Model also includes echo (I_{dte}), which we omit as this thesis focuses on pure network impacts on quality. Therefore we assume a well-working echo canceler, so that talker echo does not lead to quality degradations. Further, note that the E-Model includes other impairment factors that are not used in this thesis. The E-Model is referred to as *parametric model* since z only depends on parameters like Δ . It is also referred to as a *no-reference model*, as it does not depend on x .

Loss and Jitter effects. To assess the speech quality of y , relative to the error-free sample signal, we use the Perceptual Speech Quality Measure (PESQ) [9] model for the 'Human' subsystem from Figure 3.1. PESQ is an example of a *signal-based*

| MOS (z) | PSNR | SSIM | VQM |
|---------------|---------------------------|---------------------------|-------------------------|
| 5 (excellent) | ≥ 45 | > 0.99 | < 0.2 |
| 4 (good) | $\geq 33 \ \& \ < 45$ | $\geq 0.95 \ \& \ < 0.99$ | $\geq 0.2 \ \& \ < 0.4$ |
| 3 (fair) | $\geq 27.4 \ \& \ < 33$ | $\geq 0.88 \ \& \ < 0.95$ | $\geq 0.4 \ \& \ < 0.6$ |
| 2 (poor) | $\geq 18.7 \ \& \ < 27.4$ | $\geq 0.5 \ \& \ < 0.88$ | $\geq 0.6 \ \& \ < 0.8$ |
| 1 (bad) | < 18.7 | < 0.5 | > 0.8 |

Table 3.2: Mapping of video quality metrics to MOS scores [201, 256]

model. It takes as input both the error-free signal x and the perturbed signal y , and computes the score z_l .

The PESQ algorithm uses as input both the source voice signal x and the output perturbed voice signal y . As such, the PESQ algorithm roughly corresponds to the shortcut for x in Figure 3.1. The output of the PESQ algorithm is a score z in the range of $[1, 5]$, where 5 represents excellent speech quality. Note that PESQ does not consider Δ . It is referred to as a *full-reference* algorithm, as z_l depends on both x and y .

Overall score. The range of the score z_l , which captures loss and jitter, is $[1, 5]$. We remap it to $[0, 100]$ according to [239]. The range of the score z_d , capturing the delay impairment, is $[0, 100]$. Note, the semantics of z_l and z_d are reversed: a large value for z_l reflects an excellent quality; however, a large value for z_d reflects a bad quality, and vice-versa. We combine the two scores to an overall one as follows: $z = \max\{0, z_l - z_d\}$. Thus, if z_l is good (i.e., due to negligible loss and jitter), but the z_d is bad (i.e., due to large delays), then the overall score z is low, reflecting a poor quality and vice-versa. Finally, we map z to the MOS scale $[1, 5]$ according to the ITU-R recommendation P.862.2. In the end, low values correspond to bad quality and high values to excellent quality.

3.3.2 Video QoE for UDP/RTP Streaming

Next, we move to video streaming. More specifically, we explore the quality of video streaming based on the Real-time Transport Protocol (RTP). This protocol combination is commonly used by IPTV service providers. RTP streaming can again be impaired by packet loss, jitter, and/or delay. Packet losses directly degrades the video, as standard RTP-based video streaming *typically* does not involve any means of error recovery.⁴ Network jitter and delays result in similar impairments as in the case of voice and include visual artifacts or jerky playback. However, they depend on the concrete error concealment strategy applied by the video decoder.

⁴We investigate the QoE of RTP video streaming with additional error recovery extensions in Chapter 7.

We focus this discussion on common full-reference metrics to compute the QoE score z from the streams x (original) and y (received, perturbed signal). The output of each metric can be mapped to a quality score according to Table 3.2.⁵

For a no-reference parametric metric that captures the QoE of videos encoded in the latest video encoding standard H.264⁶, we refer to [36] that has recently resulted in the ITU-T standard P.1201.2 [21].

Note that the selection of metrics is tailored to the evaluation of non-interactive video streaming. As such, we focus on video streaming in the Internet and in IPTV networks. We exclude video conferencing that has similar delay impairment factors as discussed for speech QoE (see Section 3.3.1).

Peak Signal-to-Noise Ratio (PSNR)

The *peak signal-to-noise ratio* (PSNR) [151, 186] is a common Full-Reference *image* quality metric frequently used in image and video processing. It is based on the classical Signal to Noise Ratio (SNR) definition used in engineering, which expresses the power ratio in decibel between a signal and background noise. In image processing, the SNR measure has been adapted to describe the maximum possible signal power to corrupting noise, expressed by the Mean Square Error. The PSNR is widely used—despite of being an image quality metric even for video quality assessment—mainly due to its simple implementation in software and thus widespread availability in evaluation toolkits.

The PSNR is a function of the Mean Squared Error (MSE), which is computed as follows.

$$\text{MSE} = \frac{1}{M \cdot N} \sum_{i=1}^M \sum_{j=1}^N (x(i, j) - y(i, j))^2, \quad (3.3)$$

where $M \times N$ is the size of the image and $x(i, j)$ ($y(i, j)$) denotes the pixel value at position i, j within the original (perturbed) image. The PSNR is then defined as

$$\text{PSNR} = 10 \cdot \log_{10} \frac{\text{MAX}_I^2}{\text{MSE}} = 20 \cdot \log_{10} \frac{\text{MAX}_I}{\sqrt{\text{MSE}}}. \quad (3.4)$$

Note that the computed PSNR estimate represents a SNR measure expressed in decibel, not quality. It is thus not a QoE metric as such. The PSNR measure can, however, be mapped to a *quality* score by using mappings established in subjective tests for image quality [256] (see Table 3.2 for such a mapping).

⁵Note that the depicted quality mapping for the PSNR and SSIM image quality metrics were obtained for image quality (SSIM) and specific resolution tests (PSNR), which can question their validity to express video quality. PSNR, however, allows ranking content according to quality (see Section 3.3.2).

⁶We remark that H.265 (High Efficiency Video Coding) as the successor of H.264 is currently being standardized.

While the MSE/PSNR is appreciated for its simplicity and clear physical meaning [255], its use in quality assessment imposes limits. As its major limitation, PSNR has been shown not to correlate well with human perception [273, 191], and, has not been designed for video quality assessment. Despite of these limitations, it enables the ranking of the same video content subject to different impairments [135, 244, 156] and thus has valid use cases in video quality assessment.

Structural Similarity Index Metric (SSIM)

The Structural Similarity Index Metric (SSIM) [256] introduced by Wang et al. is motivated by the assumption that human visual perception is highly sensitive to structural information. Similar to PSNR, the SSIM metric has been designed for image quality assessment but is also widely used for video quality assessment. The SSIM score is computed based on the first and second order statistics of luminance deviations in image patches. It has been shown to have a high correlation with image quality [256], and, despite of being an image quality metric, with video quality [258, 227]. Compared to compression artifacts, the correlation with user perception has been reported to be slightly lower for errors induced by packet losses [227]. However, as compared to PSNR, SSIM can achieve higher correlations with human perception [227].

The classical SSIM metric has been extended by its original author to a Multi-Scale Structural SIMilarity (MS-SSIM) [259] and to a Speed SSIM (S-SSIM) [257] metric that are layered on top of SSIM. Subjective tests with JPEG images show slightly higher correlation with human quality perception at the cost of higher computational complexity for MS-SSIM [259].

Similar to PSNR, the SSIM score that can be mapped to a *quality* score based on mappings obtained in subjective tests (see Table 3.2).

Video Quality Metric (VQM)

The Video Quality Metric (VQM) [201] is a full-reference metric specifically tailored to assess digital video transmissions. Validation tests by the International Video Quality Expert's Group (VQEG) showed a very high correlation between subjective user surveys and the objectively measured results. Also evaluations by Seshadrinathan et al. [227] show high correlations with user perception that outperform SSIM and PSNR. This holds even in the presence of packet loss, however, with weaker correlation. The VQM method is adopted in the international ITU Recommendations ITU-T J.144 and ITU-BT.1683 since 2004.

Despite the good correlation with human perception, VQM largely differs from PSNR and SSIM in terms of runtime complexity. The much higher runtime complexity of VQM, compared to PSNR and SSIM, renders the evaluation of a large set

of videos infeasible. We therefore rely on SSIM whenever the runtime complexity induced by VQM is too large.

3.3.3 Video QoE for TCP/HTTP Streaming

We next focus on the QoE evaluation of HTTP video streaming, e.g., as used by YouTube. In contrast to IPTV and other UDP/RTP based streaming services, HTTP-based streaming uses TCP for reliable data delivery. As a consequence, neither video nor audio suffer from impairments caused by lost data. Impairments are solely introduced by the application level buffer running empty in situations when the incoming bitrate sustains to be lower than the playback bitrate. This results in rebuffering events, which alter the temporal structure of the video and result in stalling events—noticeable by jerky playback—that degrade the QoE.

While a large body of work focused on the evaluation of UDP based video streaming (e.g., see Table 3.1), a complete QoE model for TCP based video streaming is still missing. However, initial models and QoE *indicators* that contribute to the QoE of HTTP based video streaming are provided in [124, 179, 128].

The two main QoE indicators of HTTP video streaming are the number of stalling events and their respective lengths [124, 179]. The presented results suggest that users are more sensitive to interruptions. In particular when the number of interruptions dominates over the stalling length [128]. Also, users are highly dissatisfied when two or more stalling events are present [128].

Another QoE indicator is the initial buffering time denoting the time to fill up the application buffer and to start the playback. This is, however, less relevant, as users prefer waiting longer for the video to start rather than watching a jerky playback [125].

To fit our QoE model from Figure 3.1, each of the video clips corresponds to the signal x . The perturbed signal y corresponds to the video rendered by the HTTP-based video player at the end host, e.g., the YouTube player. Unlike in the context of the other QoE evaluations, the QoE literature does not provide a *general* mapping between y and a QoE score z . Therefore, for the QoE evaluation of y , we will restrict our attention to the two main QoE indicators for HTTP video streaming: stalling events and stalling times.

3.3.4 Web QoE

The web browsing experience (Web-QoE) can be related to two main indicators [71]. One is the *page loading time* (PLT), which is defined as the difference between a Web page request time and the completion time of rendering the Web page in a browser. Another indicator is the time for the first visual sign of progress [67]. In

this thesis, we consider PLT, for which an ITU QoE model (i.e., G.1030 [11]) exists to map page loading times to user scores.

To evaluate the Web-QoE, we map the PLT Δ to a user score z by using the ITU recommendation G.1030 [11]. We consider the one-page version of the ITU model, which logarithmically maps *single* PLT's Δ to scores in the range $z \in [1, 5]$ (i.e., 5:excellent, 4:good, 3:fair, 2:poor, 1:bad).

The Web-QoE for single page views can be computed as follows.

$$z = \text{MOS}_{1\text{-page}} = \frac{4}{\ln \min \Delta / \max \Delta} \times (\ln \Delta - \ln \min \Delta) + 5, \quad (3.5)$$

where $\min \Delta$ denotes the minimal page loading time and $\max \Delta$ the maximal acceptable page loading time.

3.4 Long-Term QoE Integration

We have, so far, discussed QoE estimations performed on rather short time-scales, e.g., single web page retrievals or short speech samples. The integration of multiple QoE estimates over time poses the interesting question of what users will remember from single periods of low QoE.

Concerning short time-scales spanning from several seconds up to several minutes and involving only a single interaction, research has shown that the weighting of momentary QoE judgements decays over time [260]. Thus, QoE judgements can be integrated over time by using a weighted average with higher weightings for more recent judgements. This is often described as recency effect that originates from psychological research on memory.

However, little is known about the long-term integration of QoE, i.e., the integration of QoE judgments over longer time-scales, e.g., days. Example questions concern whether users will remember single low QoE periods after multiple days. Investigating the aspect of long-term QoE integration over multiple usage periods is an emerging research topic. By evaluating Skype calls performed on a daily basis, Möller et al. [181] observed that users integrate bad QoE events into subsequent ratings within one to two days. However, the integration of these single events into final service quality rating remains unsolved. They also observed a tendency for QoE judgements to increase over time as long as the service is not interrupted. This tendency is confirmed by Guse et al. [110] for voice and audio-visual communication systems. They further find discrepancies between momentary ratings of low QoE periods and the final service quality judgement. Concretely, they find that for low-performing services, individual QoE judgments are more positive than final service quality ratings. Despite these first initial findings, the question of long-term QoE integration remains far from being answered.

3.5 Discussion

Quality of Experience aims to capture the users' perception when using a service. This is challenging since the users' perception is subjective. To address this challenge, this chapter reviewed the most common *quality* metrics and estimators for speech and video streaming, and web browsing. These metrics estimate a set of *influence factors* impacting QoE. As QoE depends on a much larger set of (subjective) influence factors (e.g., user- and context dependent factors), estimations based on currently existing QoE metrics can only be approximative and are limited to the considered influence factors. Thus, the presented quality metrics do not fully capture QoE in the strict sense and their extension will drive future research efforts.⁷

Nevertheless, the presented quality metrics provide a powerful toolbox for evaluating relevant QoE impact factors. We will instrument this toolbox in the remainder of this thesis to evaluate the QoE impact of buffering. In contrast to time intensive subjective tests, the application of objective QoE metrics allows the exploration of a large state space involving many applications, configurations, and traffic scenarios. As such approximations are naturally subject to inaccuracies, the clear next step is to verify our evaluations by using *subjective tests*. After using objective QoE metrics to reduce the complex state space to a few interesting anchor conditions, subjective tests now become feasible.

⁷We thus adopt the notion of a quality metric instead of a QoE metric in some cases.

4

QoE Management of UDP Video Streaming

In this chapter, we apply the video quality metrics discussed in the previous chapter to perform a sensitivity study on effects of packet loss, content type, content resolution and frame rate on UDP video streaming. Our first illustration highlights that packet loss has detrimental effects on video quality. Packet loss is caused by overflowing buffers that discard packets in phases of congestion. Motivated by this effect, we discuss Scalable Video Coding as control mechanisms for QoE management to optimize QoE in the presence of network congestion. Concretely, we evaluate the impact of two scalability dimensions (i.e., temporal and spatial resolution) on objective quality metrics. This preliminary evaluation serves as a motivation to broadly investigate QoE impacts of scalability in future work.

4.1 Packet Loss Impact on UDP Video Streaming

This section *illustrates* the impact of packet loss on the PSNR and SSIM quality metrics for UDP based video streaming. Packet loss mainly results from overflowing buffers and degrades the achieved quality. We remark that video transmissions can also be affected by jitter, which we will discuss in Chapter 6.

⁰The content of this chapter is joint work with Thomas Zinner, Ossama Abboud, Tobias Hoffeld, and Phuoc Tran-Gia. It has been published at ITC-SS [275] and IEEE QoMEX [274] in 2010. All experiments were executed by students at the G-Lab site at Universität Würzburg.

| Name | blue sky | crowd run | park joy |
|-----------------------------|------------|-----------|----------|
| # Frames | 216 | 499 | 499 |
| Frame rate | 30 | 30 | 30 |
| Average bandwidth (Mbyte/s) | 0.82 | 1.54 | 1.85 |
| Length (sec) | 7.2 | 16.63 | 16.63 |
| Motion type | low-medium | medium | medium |

Table 4.1: Properties of reference sequences

4.1.1 Measurement Methodology

This evaluation is based on three source videos provided by xiph.org: *blue sky*, *crowd run*, and *park joy*. Each video is available in the uncompressed y4m format at 1080p HD resolution (see Table 4.1 for further details). We encoded the raw source videos in H.264/AVC using AutoX264. The resulting H.264/AVC encoded videos represent x from Figure 3.1.

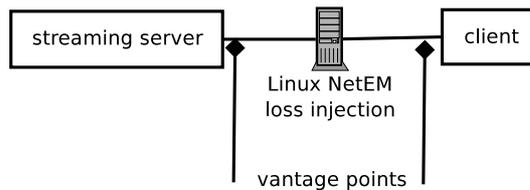


Figure 4.1: Measurement setup

We use the setup depicted in Figure 4.1 to stream the H.264/AVC encoded videos over the network. This setup represents the ‘network’ component from Figure 3.1 and uses the EvalVid framework [151] for video streaming and reassembly. To reassemble the received video stream y , both the streaming server and the streaming client are equipped with tcpdump probes that capture the outgoing / incoming packet stream. In addition, we use a host equipped with the NetEM network emulation software to inject uniform packet loss $p_l \in [0, 5\%]$. The injected packet loss corresponds to the control parameter u from Figure 3.1.

We compute the quality scores by applying the PSNR and SSIM metric to the signals x and y . As the metric is image based, we compute the overall PSNR and SSIM score by averaging the framewise scores. Note that due to the large number of samples, the computational complexity of VQM does not allow its efficient use in this evaluation.

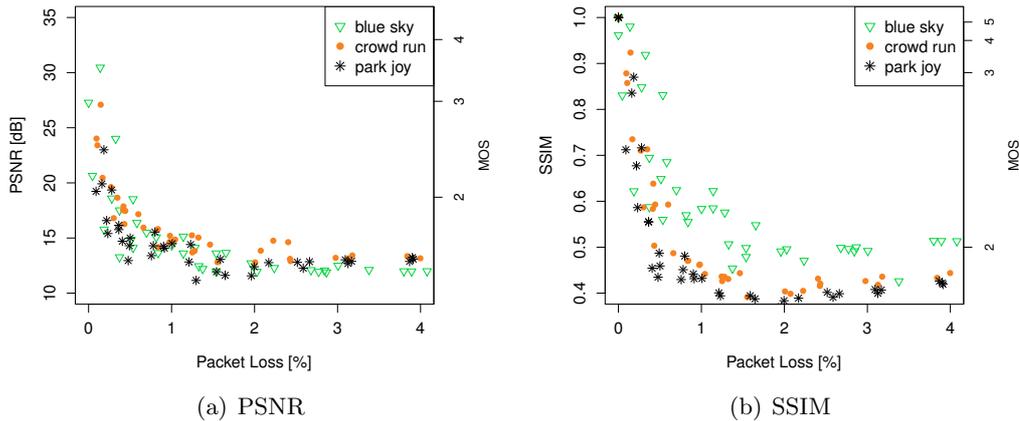


Figure 4.2: Impact of packet loss on quality scores

4.1.2 Packet Loss Influence

We next discuss the influence of different packet loss rates $p_l \in [0, 5\%]$ on quality metrics. The influence of packet loss on each of the three video contents is shown in Figure 4.2 for the PSNR and SSIM quality metric. The y-axis on the left shows the respective PSNR or SSIM score. Both, the PSNR and the SSIM score are then mapped to MOS scores according to Table 3.2 (note the exponential mapping!). The resulting MOS scores $z \in [1, 5]$ are then shown on the y-axis on the right side, where 1 denotes bad quality and 5 excellent quality.

Our first observation is that packet loss severely degrades the quality of UDP video streaming. Both metrics indicate that already a small packet loss rate significantly lowers the quality. In case of the SSIM metric, a packet loss rate $p_l = 0.007\%$ yields a MOS score of 3 (fair). Loss rates of $p_l < 0.5\%$ yield MOS scores above 2 (poor). As suggested by the IQX hypothesis (see Section 3.2), both metrics show an exponential dependency of quality as a function of the loss rate. These findings are consistent with related work (see, e.g., [215]) and highlight the fact that operators should avoid packet loss in any case.

Recent work suggests that PSNR cannot reliably assess video quality across different video contents [135]. This study considered *bit rate* changes and found PSNR to be a good indicator for *quality variations* or changes as long as the content and codec are fixed during the test. Varying contents, however, lead to unstable results. Figure 4.2(a) illustrates the PSNR estimation for different contents subject to different *loss rates*. The figure shows similar PSNR patterns for each of the investigated contents. In other words, the content type of the selected videos did not lead to clearly visible differences in the PSNR estimation.

In this evaluation, SSIM on the other hand exhibits some level of content sensitivity: the effects of packet loss on the SSIM predictions depend on the respective content.

This behavior is shown in Figure 4.2(b). The figure shows that *blue sky*—which also has the lowest motion complexity—is affected the least by packet loss. On the other hand, the *park joy* scene—which has the highest motion complexity—is affected the most. This evaluation shows that the impact of packet loss is dependent on the motion complexity: the higher the motion complexity, the larger the traffic bursts (spatio-temporal complexity), and the higher the impact of packet loss. Related work made similar observations for the PSNR metric, i.e., content with higher motion complexity is more sensitive to packet loss [147]. In contrast to this evaluation, their observations were made for higher packet loss rates $> 10\%$ and lower content quality.

This evaluation illustrated two aspects: *i*) packet loss significantly lowers the video quality as predicted by PSNR and SSIM and *ii*) content-dependent effects¹ can be more pronounced in SSIM. This does not imply that PSNR is content insensitive. Further, while these observations have been also made for human perception (see e.g., [215, 220, 126] for the impact of packet loss), the respective effect strength and characteristics as predicted by PSNR and SSIM are likely to differ from human perception. We leave a detailed study of content-dependent aspects for future work and point to an existing body of work on packet loss sensitivity (e.g., [215, 220, 126]) and temporal complexity (e.g., [15]).

4.1.3 Trade-off between Packet Loss and Content Quality

Due to its detrimental effects on the quality of UDP based video streaming, packet loss should be avoided. Assuming a congested network, one way to avoid packet loss is by reducing the bandwidth of the video stream. Bandwidth reductions can be achieved by reductions of the video resolution. This section compares packet loss degradations obtained in Section 4.1.2 to frame size reductions. Frame size reductions cause blurriness when the video is displayed in the original size. Packet loss impairments, on the other hand, result in block artifacts and can cause jerky playback or frame freezes, depending on the decoders concealment strategy. Due to the lower content sensitivity of PSNR in the previous evaluation (see Section 4.1.2), we only focus on the SSIM quality metric.

Figure 4.3 compares the impact of packet loss to frame size reductions on SSIM quality predictions. As in Section 4.1.2, the x-axis shows the random packet loss rate, and the y-axis shows the corresponding SSIM values. Packet loss degradations are only shown for the video in the highest resolution of 1216x684 pixel. In this

¹There are different types of content dependent effects. On the packet level, the error propagation and the spatial extent of a packet loss are influenced by the spatio-temporal complexity (e.g, size of I, P, and B frames depending on the motion complexity) of the video and its encoding pattern (e.g., bitrate and GOP configuration). On the visual level, the duration of error propagation is limited by the frequency of scene changes and the GOP pattern of the encoded video. Also structural pattern in the video determine the visibility of packet losses (see for instance [142, 219, 143, 36]).

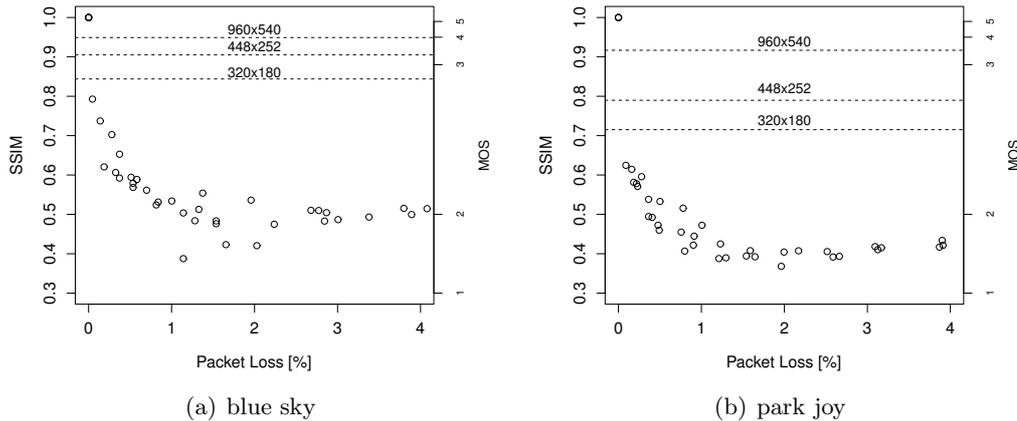


Figure 4.3: Tradeoff between packet loss and content quality

example, we assume that video streams in reduced quality settings do not experience packet loss. The dashed lines indicate the corresponding SSIM values for content in lower resolution of 960x540, 448x252, and 320x180 pixels, respectively. We used the nearest neighborhood interpolation method to rescale the videos. This yields a lower bound, as this interpolation method yields the lowest quality results compared to other interpolation methods (see Section 4.2.3).

As shown in the figure, resolution reductions still yield reasonable quality scores while saving bandwidth. In the case of the *blue sky* scene (see Figure 4.3(a)), downsizing the video to a resolution of 960x540 reduces the used bandwidth by 25% and still yields a quality score of 4 (good). Even further reductions in the video resolution reduce the used bandwidth by more than 50% and result in a quality score of 3 (fair). The SSIM scores achieved for the reduced resolution equal a packet loss rate of $p_l \leq 0.04\%$. Similar observations can be made for the *park joy* scene (see Figure 4.3(b)). As *park joy* differs from *blue sky* in terms of content, the SSIM thresholds for the various resolutions also differ. However, degradations due to packet loss rates $p_l \geq 0.01\%$ yield lower quality scores than for resolution reductions.²

4.1.4 Discussion

This section illustrated the impact of packet loss on the PSNR and SSIM quality metrics. To avoid packet loss, we discussed resolution reductions as one possible method to reduce the required bandwidth in congested networks. Concretely, this evaluation illustrated the following aspects:

²Note that the impact of packet loss on the *park joy* is more severe due to the higher motion complexity, see Section 4.1.2. This leads to different packet loss thresholds.

- Even low amounts of packet loss have detrimental effects on UDP video streaming and should thus be avoided.
- Content-dependent packet loss effects are more pronounced in SSIM.
- Resolution and thus bandwidth reductions yield better PSNR and SSIM quality scores than even low amounts of packet loss.

These observations suggest that in case of network congestion, bandwidth reductions yield better quality scores than in the presence of packet loss. This motivates the study of bandwidth reduction mechanisms in Section 4.2.

We note that our observations are entirely based quality predictions by common approximative quality metrics. Human perception might differ from the above presented assessment. We leave this investigation to future work.

4.2 Bandwidth Reduction by Scalable Video Encoding

As packet loss has detrimental effects on the quality of UDP based video streaming, we next evaluate bandwidth reduction mechanisms. Bandwidth reduction has the potential to lower the packet loss rate when the network is congested. In general, without changing encoding settings such as the encoding bitrate, the bandwidth required for video streaming can be reduced by either reducing the spatial resolution (i.e., the frame size), or by reducing the temporal resolution (i.e., the frame rate). Spatial reductions will lead to blurriness when the video is viewed full screen. Temporal reductions can cause jerky playback. This study is motivated by evaluating trade-offs between quality reductions due to temporal or spatial reductions and packet loss.

We focus on Scalable Video Encoding (SVC) that allows seamless quality switches during video streaming. SVC was specifically designed to adapt the video streaming quality to current network conditions. Concretely, we use the SVC extension of H.264—a video coded that is widely used in the Internet, e.g., by YouTube or Zattoo. The focus of this section is to evaluate the impact of different SVC scalability dimensions (i.e., frame rate reductions and resolution reduction) on common quality metrics (i.e., SSIM and VQM). Such quality metrics were not specifically designed to evaluate such scalability effects and their quality estimates are thus likely to differ from human perception. Nevertheless, this evaluation highlights the impact of partial-scalability changes in SVC on common full reference quality metrics. We leave the respective evaluation of human perception for future work.

4.2.1 Scalable Video Encoding

The SVC extension enables the encoding of a video at different qualities within the same layered bitstream. Besides different resolutions, this also includes different temporal layers (frames per second) and quality layers. These three dimensions are denoted as spatial, temporal, and quality scalability.

The scalable video stream can be viewed in three different temporal resolutions (e.g., 15 Hz, 30 Hz, 60 Hz), three different spatial resolutions (e.g., CIF, SD, HD), and three different quality resolutions (i.e., Q0, Q1, Q2). The base layer has minimal quality settings and is the only layer required for video playback. Additional enhancement layers improve the quality and thus the quality. The video can be viewed in the best possible quality when all enhancement layers are transmitted.

SVC allows to seamlessly switch between quality settings by controlling the number of streamed enhancement layers. This feature allows an adaptation of the video quality to service parameters. In the case of congestion, the bandwidth requirements can be reduced by transmitting less enhancement layers, i.e., reducing the resolution, frame rate, or quality.

4.2.2 Measurement Setup

Our study is based on the same set of videos as used in Section 4.1. We encoded the raw source videos in H.264/SVC using the JSVM software version 9.15. For the SVC encoding, we used different spatial and temporal layers. The base layer comprises a resolution of 480x270 pixels with a frame rate of 1.875 frames per second (fps). The base layer is then extended by several spatial and temporal extension layers. We encoded the temporal extension layers at four frame rates of 3.75, 7.5, 15, and 30 fps. Further, we encoded three spatial enhancement layers with resolutions of 640x360, 960x540, and 1216x684 pixels. The maximum video quality of 1216x684 pixels at 30 fps is achieved when all layers are available.

To compare videos in lower resolutions to the maximum quality, we scaled videos in lower resolution up to the highest possible resolution. This represents a situation in which a user prefers to watch videos full screen, despite of their respective resolution. In this process, we compare two different scaling methods: nearest neighborhood and bicubic interpolation. As the SSIM and VQM metrics can only evaluate videos that have the same number of frames as the 30 fps reference video, we also had to scale up the temporal resolution of the low fps videos. We scaled the temporal resolution by replacing missing frames with their predecessors.

As in Section 4.1, we compute the quality score by applying the SSIM metric. We exclude PSNR from this evaluation as it has been shown to not to correlate well with human perception [273, 191] and to be less sensitive to content dependent effects as shown in Section 4.1. Also, related work studied the use of PSNR in the context

of SVC [166], which lets us concentrate on more advanced metrics. We additionally include the VQM metric due to its better correlation with human perception (see Section 3.3.2). As the number of test cases is smaller than in Section 4.1, the computation of VQM is feasible.

4.2.3 Video Resolution

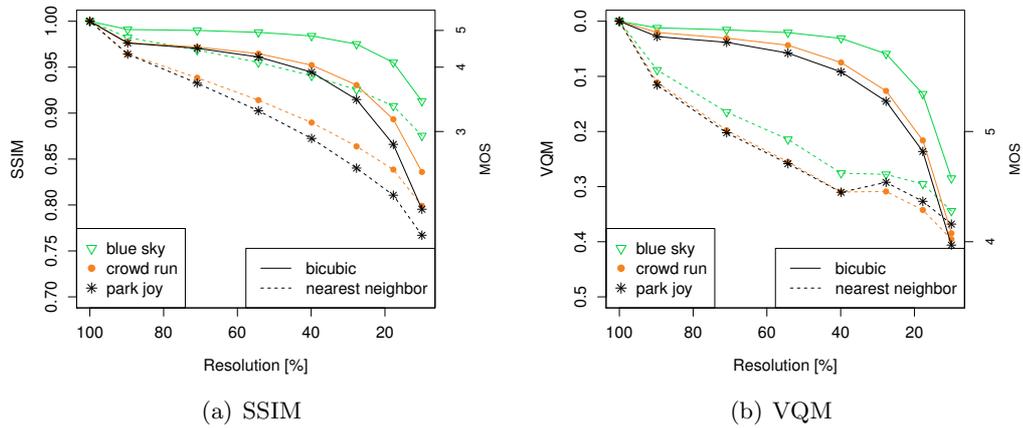


Figure 4.4: Quality impact of video resolution

We first discuss the influence of resolution reductions on quality metrics. In contrast to Section 4.1.3, this evaluation is detached from packet loss impairments and entirely focuses on control mechanisms provided by SVC based video streaming.

Our results are shown in Figure 4.2.3 for two quality metrics SSIM and VQM. In both figures, the x-axis shows the resolution reduction. A value of 100% represents unchanged quality and represents the baseline for our evaluation. Assume the quality to be optimal in this setting. We evaluate resolution changes with respect to this baseline quality. The quality score computed by the respective metric is shown on the y-axis. We map this score to a perceptual MOS score according to Table 3.2 and show it on the y axis on the right side.

In a second step, we evaluate two different interpolation mechanisms used to reduce the video resolution, i.e., nearest neighborhood and bicubic interpolation. The quality for each interpolation method is shown as solid or dashed lines. The dashed lines show the results for nearest neighborhood interpolation, whereas the solid lines show the results for bicubic interpolation.

In general, resolution reductions yield quality reductions. This is shown in Figure 4.4(a) for the SSIM metric and in Figure 4.4(b) for the VQM metric. For both metrics, a decrease in resolution yields a decrease in quality, regardless of the respective content.

Further, it can be seen, that nearest neighborhood interpolation always yields lower quality scores than bicubic interpolation. Despite its higher computational complexity, bicubic interpolation should be the preferred method when resizing video streams.

The VQM metric differs from the SSIM metric in two points. First, VQM is not sensitive to the used interpolation method for a small resolution of 384x216 pixels. In these cases, the computed MOS score is equally low. Further, VQM yields higher quality scores. Concretely, SSIM assesses low resolutions with MOS values ranging from poor (2) to fair (3). In contrast, VQM yields good (4) quality scores for the same resolution. For high resolutions, however, the differences between both metrics are not as significant.

This evaluation showed that both objective metrics are sensitive to resolution changes. However, both methods differ in the predicted quality, mostly when assessing influences of interpolation methods. Thus, a subjective assessment of the investigated scenarios is needed to link the objective metrics to the real user perceived quality. Similar to the influence of packet loss, content aspects influence the obtained quality scores.

4.2.4 Frame Rates

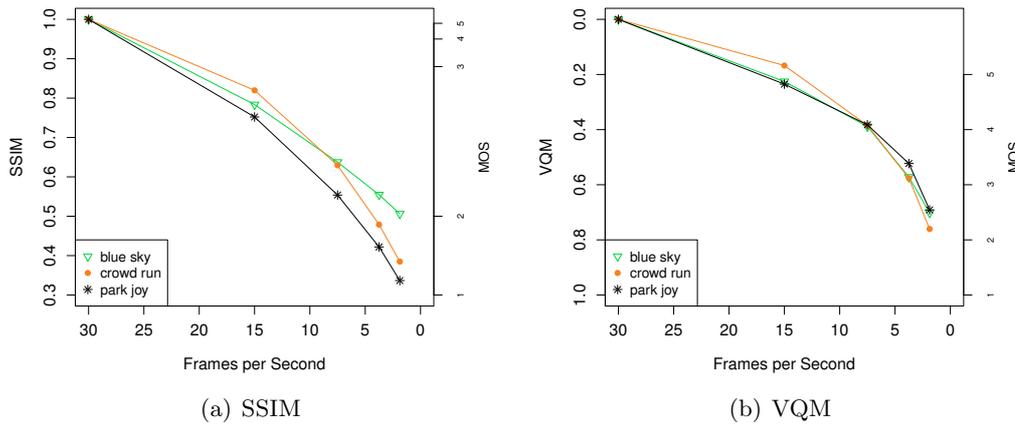


Figure 4.5: Quality impact of different frame rates

Another method to reduce the bandwidth of video streams is to reduce the frame rate (temporal scalability). The quality degradations for frame rate reductions are shown in Figure 4.5, again for the two quality metrics SSIM and VQM. The frame rate is shown on the x-axis. As in the previous section, the resulting quality score is shown on the y-axes as SSIM/VQM value (left axis) and MOS score (right axis).

In general, reducing the frame rate reduces the quality. This trend has been observed in subjective tests [134] and is shown in Figure 4.5(a) for the objective SSIM metric

and in Figure 4.5(b) for the objective VQM estimate. However, as in Section 4.2.3, the VQM metric yields higher quality scores than SSIM. For example, reducing the frame rate to 15 frames per second yields to MOS value fair (3) for SSIM and a MOS value of good (4) or better in case of VQM.

As for the previous section, we note that the reported quality impacts are solely based on approximative quality metrics (i.e., SSIM and VQM). The shown quality impacts due to frame rate changes (i.e., jerkiness) are thus subject to SSIM and VQM quality estimations. Human perception is likely to differ from these predictions. In particular, as SSIM computes quality scores based on pixel-based measures on the perturbed and reference video and is not designed for evaluating frame rate changes. VQM on the other hand considers spatial and temporal impairments.

4.2.5 Trade-Off between Bandwidth, Frame Rate and Resolution Reduction

Having discussed bandwidth reductions by frame rate and resolution reductions separately, we next jointly discuss the trade-offs between the different optimizations. We show the trade-off between bandwidth savings by either resolution or frame rate reductions in Figure 4.6, 4.7, and 4.8 for the three investigated video sequences. In all figures, the x-axis shows bandwidth savings subject to the maximum quality. The y-axis displays the VQM/SSIM quality score (left side) and the MOS score (right side).

In all of our evaluations, SSIM/VQM quality degradations due to frame rate reductions are higher than for resolution reductions. To optimize SSIM and VQM quality scores, bandwidth reductions should be achieved by resolution changes rather than frame rate changes. In particular, frame rate reductions yield less bandwidth savings and lower SSIM/VQM quality scores.

We again note that our assessment is purely based on scores computed by quality metrics. Subjective quality might differ from SSIM and VQM scores.

4.3 QoE Management

QoE management provides the ability to adapt video streaming to changing network conditions while optimizing the overall QoE. Changing network conditions—e.g., reduced available bandwidth during congestion—lead to quality reductions. Such quality reductions can be caused by two major factors.

1. **Packet loss.** Even low amounts of packet loss can significantly degrade the overall quality by causing visual artifacts and/or jerky playback (see Section 4.1).

2. **Stalling events.** Jitter and insufficient available bandwidth causing losses and delays can lead to jerky playback that can also significantly degrade quality (see e.g., [125]).

We argue that the above mentioned problems can be addressed by bandwidth adaptations. Such adaptations can be achieved by either *i*) reducing the video resolution (spatial reduction), *ii*) decreasing the image quality (quality reduction), or *iii*) reducing the frame rate (temporal reduction). In this setting, SVC provides a powerful tool that is specifically designed for such use cases.

In a measurement study, we evaluated the impact of bandwidth adaptations to objective quality metrics. We particularly focused on resolution reductions with different scaling methods and frame rate reductions (partial scalability). We showed that the SSIM and VQM QoE metrics can distinguish between quality levels for different resolutions and frame rates.

Our evaluation leads us to discuss a QoE management scheme based on partial scalability. In this scheme, bandwidth adaptations are preferably achieved by spatial resolution reductions rather than frame rate reductions. The latter showed less bandwidth savings and lower QoE scores than resolution changes. After conducting our study, similar observations have been confirmed in a subjective study by Lee et al. [165] for high bitrates. Only in low bit rate conditions, temporal reductions are preferred over spatial reductions. Future work should therefore extend this first step and investigate QoE management approaches that were evaluated in subjective studies.

4.4 Related Work

4.4.1 Prior Work

Recall that bandwidth reduction is usually achieved by one of the following ways *i*) image resolution reduction, *ii*) image quality degradation by increasing the compression rates (larger quantization), or *iii*) frame rate (fps) reductions. User studies investigated QoE influences by these influence factors, especially in the context of mobile environments. In [53] Buchinger et al. described the interconnection between the compression rate and the frame rate in mobile environments. This study shows that users prefer spatial quality over temporal quality (frame rate). Similar observations have been made by McCarthy et al. [177]. They showed low resolution videos on desktop computers and palmtops to subjects. In particular, they used a resolution of 352x244 pixel for the desktop experiments, and 176x144 pixel for the palmtop experiments. These surveys confirm, that users tend to neglect a reduction of the frame rate. Decreases in picture quality, however, results in dissatisfied users. Knoche [153] pointed out that users prefer to watch video sequences at the largest

possible size. However, he did not investigate the QoE impact of scaling mechanisms that we study in Section 4.4.

Many studies investigated the impact of packet loss on video quality. Raake et al. [215] provides a parametric no-reference model to investigate the quality of H.264 based UDP video streaming services. Khan et al. [147] studied the impact of high packet loss rates $> 10\%$ on the PSNR metric and identified content dependent factors and acceptable bitrate ranges. In contrast to their work, we study the impact of packet loss on a larger set of quality metrics, by using video content in a higher resolution, and by focusing on more modest packet loss rates. Greengrass et al. [106] investigates the impact of packet loss durations on QoE. Boulos et al. [50] also studied transmission related factors such as the loss distribution. Their study also investigated content-related factors such as the scene-cut position and find the impact of packet loss to be content-dependent. The visibility of packet losses is investigated by Kanumuri and Reibman [142, 219, 143, 144].

This evaluation differs from related evaluations by *jointly* evaluating the quality influence of packet loss, content, resolution, and frame rate on the same set of video sequences / contents. While these aspects have been studied in isolation on different contents, the content sensitivity challenges the comparison among previous studies.

4.4.2 Follow-up Work

When we investigated QoE management, adaptive video streaming was an emerging topic that is now becoming widespread. In particular HTTP-based video streaming technologies, e.g., as used by Netflix, have widely adopted this idea. Adaptive HTTP based video streaming is offered by many commercial products, including Adobe dynamic streaming for Flash, Apple adaptive streaming, and Microsoft Smooth Streaming. The development of Dynamic Adaptive Streaming over HTTP (DASH) as open standard started after our study in 2010. It resulted in an ISO/IEC MPEG and 3GPP standard [185] in 2011. One year later, QoE-aware DASH (QDASH) [180] has been proposed. QDASH adapts the streaming quality to the measured available bandwidth. In addition, Sieber et al. [229] evaluate user-centric approaches for quality adaptation in DASH, highlighting that quality adaptation still is a relevant research direction. Akhshabi et al. [32] provides an overview of current adaptive streaming protocols.

Similarly, scientific studies complemented our work by focussing on subjective evaluations of Scalable Video Coding. We specifically discussed the need for such studies in future work. Based on subjective studies, Ukhanova et al. [249] proposed a PSNR-based method to derive the impact of frame rate reductions. Lee et al. [165] investigated SVC-based QoE management in a subjective study. Their study shows that high frame rates is preferred over spatial resolution in high bit rate settings,

which is in line with our findings. Contrary to our work, they find that spatial resolution is preferred over frame rates in low bitrate settings. An overview of quality assessment in the context of SVC is provided in [166].

Other works have extended the presented idea of QoE management by partial scalability. Abboud et al. [25] extended our SVC based concept to P2P video streaming. Li et al. [169] extends QoE management based on partial scalability to full scalability. Similarly, Li et al. [168] investigates SVC to optimize QoE in wireless networks.

4.5 Discussion

This chapter started with an illustration of quality degradations on PSNR and SSIM scores caused by packet loss. This illustration highlighted that that packet loss has detrimental effects on quality and should therefore be avoided. Packet loss can be avoided in congested networks by reducing the bandwidth footprint of the video stream. Bandwidth reductions can be achieved by either *i*) video resolution reductions, *ii*) bit rate reductions, or *iii*) frame rate reductions.

These reductions can be deployed by using Scalable Video Encoding (SVC). SVC is a powerful tool to optimize video quality by seamless bandwidth adaptations. The performed evaluation of bandwidth adaptation mechanisms showed that resolution changes yield better overall SSIM/VQM quality scores and higher bandwidth savings than frame rate reductions.

Note that this evaluation is entirely based on objective quality metrics. As such, subjective user perception is likely to differ from the shown objective assessments. In particular, as for example SSIM was not designed for evaluating frame rate reductions. However, studies that were conducted after our work confirm our findings for at least high bitrate settings.

4.6 Future work

In the context of video delivery in the Internet, we can see two emerging trends.

1. Video streaming in the public Internet (over the top services) will dominantly use HTTP and open protocols. As the Internet as a whole is designed as best effort network and is not QoS enabled (see Section 2.6), adaptive streaming using DASH as an open standard will become dominant and help to optimize QoE in the absence of QoS and overprovisioned networks. Further understanding QoE implications and optimizations of DASH will therefore become an emerging topic.

2. UDP based video streaming will be used for IPTV / Video on Demand services offered by ISPs (managed services, e.g., ISP-internal IPTV). As this traffic does not traverse the public Internet, it can easily be QoS enabled. We thus see a declining need for RTP-based SVC.

QoE research has largely focused on the assessment of Voice over IP and UDP based video streaming. Starting in 2011, QoE research has slowly investigated the QoE of HTTP based video streaming, that is increasingly popular in the current Internet. These research efforts have, however, not yet resulted in a complete QoE model. Similarly, a good model to capture web QoE is still missing. Due to the increasing usage of HTTP-based video streaming, future work should therefore focus on providing QoE models for HTTP video streaming.

The presence of meta content distribution networks (CDN) will strengthen the need for objective QoE metrics, in particular for HTTP based video streaming. Such networks select the CDN from which the content is being served by objective criteria. One of such criteria is video quality. This approach has been discussed in [171]. In order to optimize QoE delivery, meta network operators need powerful objective QoE metrics that optimize QoE by choosing from a selection of CDNs.

Also, QoE management by using adaptive video streaming has so far only considered edge based control by the sender. A more realistic and relevant scenario is QoE management run in the core. In this setting, a QoE and application aware router would drop SVC enhancement layers to reduce the load in the network.

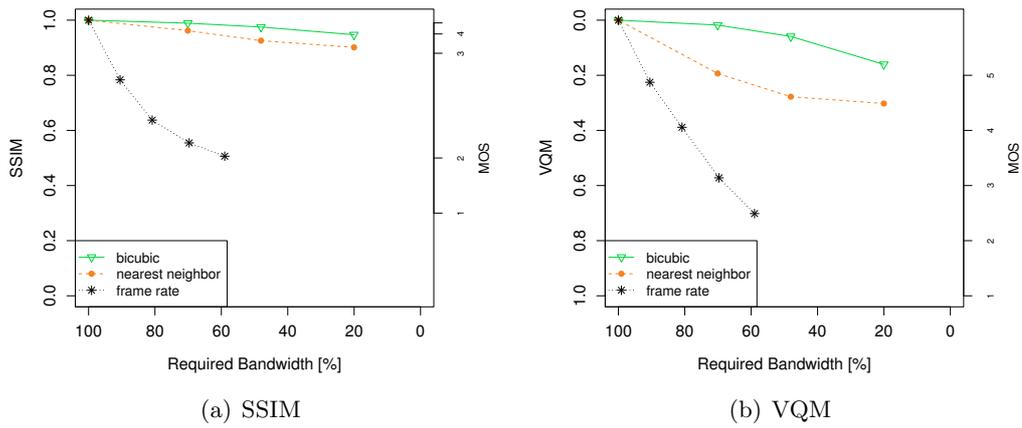


Figure 4.6: Video clip blue sky

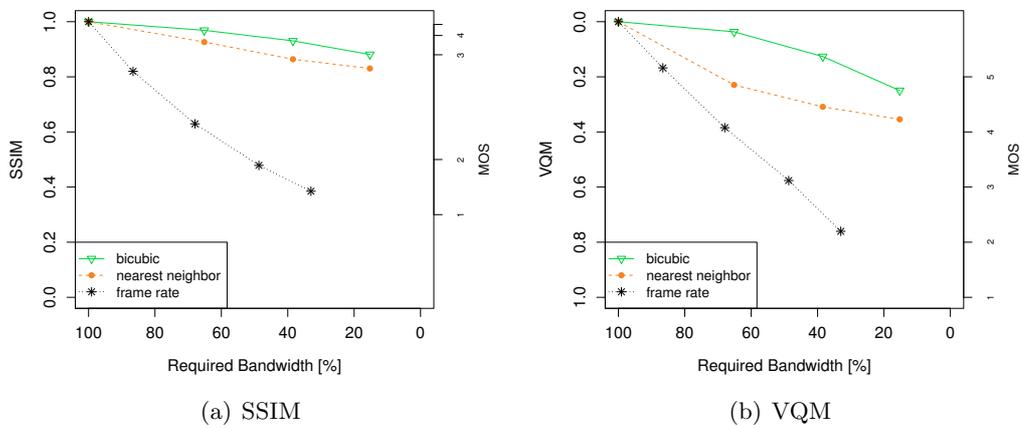


Figure 4.7: Video clip crowd run

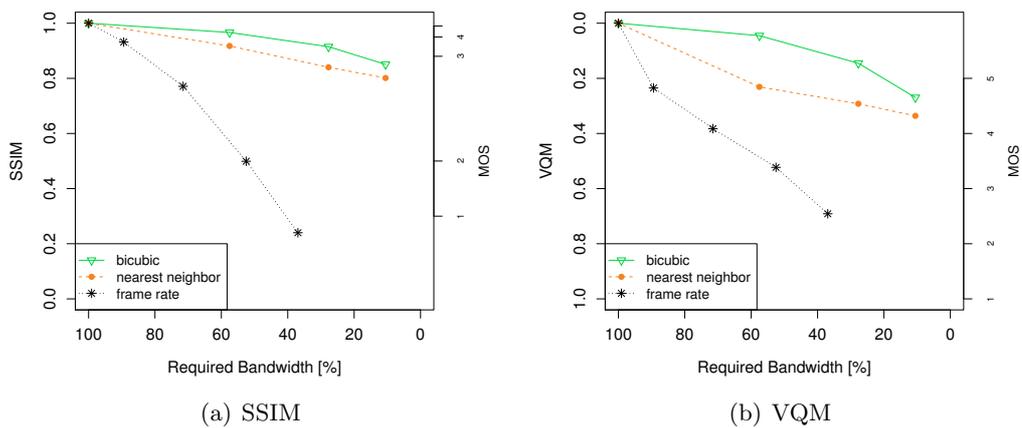


Figure 4.8: Video clip park joy

Part I

Impact of Network Buffer

5

QoE Impact of Packet Loss Models

Packet buffers in routers, switches, and hosts do not only introduce (queueing) delay, but also drop packets once their capacity is exceeded. As shown in the previous chapter, the resulting packet loss can cause quality degradations, in particular in the case of real-time video transmissions over UDP. This quality impact of packet loss can be investigated by sending video streams through testbeds and then deriving visual quality properties by decoding the resulting sequences. This is the approach used in the previous chapter. It is beneficial, as it accurately reflects the state of the network. On the other hand, the complexity of real networks challenges the generation of loss processes with desired loss rates and burst length distributions.

Such a fine-grained control over the loss process is required for subjective tests used to construct QoE models. In these tests, videos are perturbed with different loss patterns (e.g., loss rates and loss burst lengths). The resulting model then allows a mapping between loss patterns and QoE. To generate controlled loss processes, such studies often rely on model-based generators rather than realistic networks / testbeds (see e.g., [215] for a QoE test based on Markovian loss generators). The flexibility of model-based loss generators permit studying the behavior of video sequences impaired under different error conditions. A disadvantage of model-generated loss traces is that the statistical properties may not adequately replicate those of the measured trace, as they are likely to be biased by model limitations. This chapter

⁰The content of this chapter is joint work with Florin Ciucu and has been published in ACM SIGMETRICS MAMA [119]. This chapter extends the fitting scheme presented in the authors master thesis [117] by an evaluation of model based QoE impacts. The fitting scheme is joint work with Gerhard Haßlinger.

aims at studying how the loss model choice biases visual quality properties and thus QoE estimates.

Finite-state Markov chains are a particular class of models for generating packet loss traces. In general, these models are widely used to characterize error processes in telecommunication systems and to evaluate the performance of coding or other measures for error resilience [216, 231, 247]. A large body of work focuses on replicating loss burst and gap length distributions, which are relevant for evaluating the performance of FEC algorithms.

In the case of encoded video, however, errors at various time-scales may cause different visual effects, e.g., distorted areas of an image. Depending on the bitrate, time-scales in video can range over multiple orders of magnitude from microseconds in the case of the transmission of a single macroblock—a small part of a video frame—up to seconds for the transmission of an entire group of pictures. Moreover, with the advent of the latest video encoding standard H.264 [192], parameter sets describing a set of pictures with similar decoding properties can introduce even larger time-scales. While losing a single macroblock can have a minor impact by affecting only a small fraction of the image, losing a parameter set can have a severe effect on the decoding process. This consideration suggests to focus on the loss process over multiple time-scales rather than on matching burst and gap length distributions.

To address this observation, we describe a new fitting technique that we first introduced in [117]. This technique is optimized for replicating error patterns in multiple time-scales. This technique fits an M -state Markov chain to loss processes observed in measurements. The fitted Markov model is then used to generate loss traces for impairing video transmissions. We compare the proposed techniques to classical Markov fitting approaches.

In contrast to our earlier work, this chapter focuses on evaluating model impacts on video quality. By using moment matching, 2-state Markov models are fitted to a wireless Digital Video Broadcasting (DVB-H) loss trace. The fitted models are then used to generate packet losses that impair video sequences. Instructing a video decoder lets us derive visual error patterns that result from packet losses. By looking at visual error patterns, we take a microscopic view at factors influencing QoE. Our findings show that error models alter these microscopic aspects and suggest that loss generators should thus be carefully chosen.

5.1 Markovian Loss Models

A discrete Markov chain with a set of M states $S = \{1, \dots, M\}$ characterizes the course of the process with regard to the current state, which may change over time at packet arrivals, based on transition probabilities. Each state is associated with different packet loss behavior. Let q_t denote the current state at event time $t \in \mathbb{N}_0$.

Then the probabilities a_{ij} to change from state $q_{t-1} = i$ to $q_t = j$, $i, j \in S$, are given in the transition matrix A , with coefficients

$$a_{ij} = P(q_t = j | q_{t-1} = i), \quad i \leq i, j \leq M, a_{ij} \geq 0; \quad \sum_{j=1}^N a_{ij} = 1.$$

We restrict our considerations to homogeneous, irreducible, and aperiodic Markov chains. The model guarantees the existence of steady state probabilities π_k . Finally, we define packet loss rates in each state $E = (e_1, \dots, e_M)$; $0 \leq e_j \leq 1$ as the probability that state i will drop a packet with probability e_i . This Markov process outputs the loss process $O(t)$, where $O(t) = 1$ indicates a lost packet and $O(t) = 0$ stands for error-free events, respectively.

5.2 Second-order Statistics of Loss Processes

Second-order statistics in multiple time scales are a standard approach capturing and to describing traffic variability including long-range dependencies and self-similarity [246, 167]. Following this trend and summarizing our previous work [117], we next briefly describe second-order statistics, in particular the coefficient of variation of the number of packet losses over a range of time frames.

In order to capture a packet loss process generated by an M -state Markov model, we can set up recursive equations for the distribution function of losses in a considered sequence of packets. Let $p_n^{(m)}(k)$ denote the probability of k packets being lost in a sequence of length n generated by a Markov process, which starts in steady state and finally resides at state m ($n \in \mathbb{N}_0$ and $1 \leq m \leq M$).

The probabilities $p_{n+1}^{(m)}(k)$ for sequences of length $n + 1$ can be recursively computed from $p_n^{(m)}(k)$, where steady state starting conditions are expressed as $p_0^{(m)}(0) = \pi_m$:

$$p_{n+1}^{(m)}(k) = \sum_{j=1}^M p_n^{(j)}(k) (1 - e_j) a_{jm} + p_n^{(j)}(k-1) e_j a_{jm}.$$

As a standard approach to obtain the mean and the variance of the distributions of packet losses we introduce corresponding generating functions defined as

$$L_n^{(m)}(z) \stackrel{\text{def}}{=} \sum_{k=0}^n p_n^{(m)}(k) z^k.$$

A generating function $L_n^{(m)}(z)$ comprises the probabilities $p_n^{(m)}(k)$ for $k = 0, \dots, n$. The recursive relationship for the probabilities $p_{n+1}^{(m)}(k)$ transfers into the generating function notation as follows:

$$L_{n+1}^{(m)}(z) = \sum_{j=1}^M L_n^{(j)}(z) (1 - e_j + e_j z) a_{jm}. \quad (5.1)$$

According to steady state starting conditions, we again have $L_0^{(m)}(z) = p_0^{(m)}(0) = \pi_0$. In addition, $L_n^{(m)}(1) = \sum_k p_n^{(m)}(k)$ generally holds for the sum of the probabilities of a distribution by definition of generating functions. Since the considered process stays in steady state, we have $\sum_k p_n^{(m)}(k) = \pi_m$ as the probability to find the process in state m after n packet arrivals. Therefore $L_n^{(m)}(z)$ represent defective distributions with regard to a final state m of the Markov chain, while their sum $L_n(z) = \sum_{m=1}^M L_n^{(m)}(z)$ characterizes the complete distribution of packet losses during a sequence of n packet arrivals with $L_n(1) = \sum_{m=1}^M L_n^{(m)}(1) = \sum_{m=1}^M \pi_m = 1$.

The mean values $\mu_n^{(m)}$ for the distributions $p_n^{(m)}(k)$ can be derived via the first derivative of the generating function using the rule $\mu_n^{(m)} = \frac{d}{dz} L_n^{(m)}(z)|_{z=1}$, resulting in

$$\mu_n^{(m)} = \sum_{j=1}^M \pi_j e_j \sum_{k=1}^n a_{jm}^{(k)}. \quad (5.2)$$

Next, we proceed with the second order statistics of the process to calculate the variance of the number of packet losses as our focus. Therefore, the second derivatives of the generating functions are again evaluated at $z = 1$:

$$\begin{aligned} \frac{d^2}{dz^2} L_{n+1}^{(m)}(z) &= \sum_{j=1}^M \left(\left(\frac{d^2}{dz^2} L_n^{(m)}(z) \right) (1 - e_j + e_j z) + 2 \left(\frac{d}{dz} L_n^{(m)}(z) \right) e_j \right) a_{jm} \\ \Rightarrow \nu_{n+1}^{(m)} &= \sum_{j=1}^M (\nu_n^{(j)} + 2\mu_n^{(j)} e_j) a_{jm}, \end{aligned}$$

where $\nu_n^{(m)} \stackrel{\text{def}}{=} \frac{d^2}{dz^2} L_n^{(m)}(z)|_{z=1}$. Summing up over the final state m of the Markov process, we approach the result for the complete distribution

$$\sum_{m=1}^M \nu_n^{(m)} = 2 \sum_{j=1}^M \sum_{r=1}^M \pi_j e_j e_r \sum_{k=1}^{n-1} (n-k) a_{jr}^{(k)}.$$

Using $\sum_m \mu_n^{(m)} = \sum_m \pi_m e_m$, i.e., the mean loss rate ne , we obtain the coefficient of variation

$$\begin{aligned} c_v(n) &= \frac{1}{ne} \sqrt{ne - (ne)^2 + \sum_{m=1}^M \nu_n^{(m)}} \\ &= \frac{1}{ne} \sqrt{ne - (ne)^2 + 2 \sum_{j=1}^M \sum_{r=1}^M \pi_j e_j e_r \sum_{k=1}^{n-1} (n-k) a_{jr}^{(k)}}. \end{aligned} \quad (5.3)$$

The previous result is generally applicable to compute $c_v(n)$ at a moderate computational complexity, including the determination of the k -step transition matrices for $k = 1, \dots, n$ as the main step. In addition, it is well known that the coefficients $a_{jr}^{(k)}$ can be expressed in closed form by solving the corresponding eigenvalue problem; numerically this is feasible even for large values of M [245].

5.2.1 Empirical DVB-H Loss Trace

For the loss model assessment in this section, we rely on a DVB-H error trace. The DVB-H standard describes Digital Video Broadcasting for Handhelds and was standardized by the European Telecommunications Standards Institute in 2004 [82] as standard for mobile TV broadcasting. The QoE of TV broadcasting services is impacted by transmission errors naturally occurring in error prone wireless channels. To assess the discussed error models using a realistic loss trace, we use a DVB-H packet error trace captured in the DVB-H lab at the University of Turku, Finland. We refer to [207] for a description of the measurement setup.

5.2.2 Parameter Estimation

The parameters of the Markov chain can be obtained by fitting the analytical curve $c_v(n)$ from (5.3) to the empirical curve $c_v^{\text{empirical}}(n)$ of a trace. This can be done using standard methods for numerical optimization of non-linear functions. The quality of the solution can be evaluated using the mean square error function

$$\text{MSE} = \sum_{n=1}^N \left(c_v(n) - c_v^{\text{empirical}}(n) \right)^2$$

for time scales $1, \dots, N$, where a smaller MSE for a particular model indicates a better fit.

Using the proposed fitting technique, two versions of a 2-state Markov model have been fitted to a DVB-H error traces on time-scales ranging from 1 to 10^5 packets. A simple version consists only of the two transition probabilities (the loss rate in

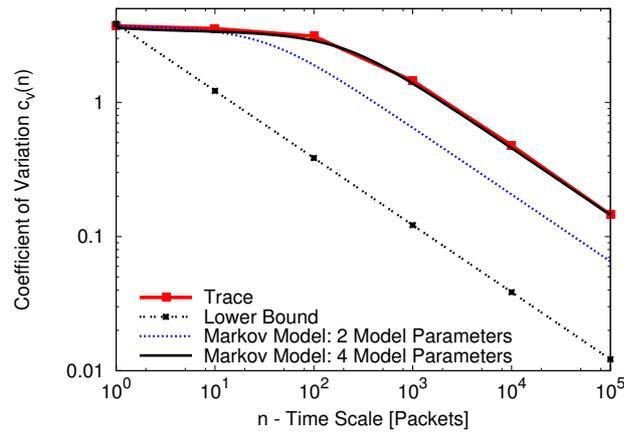


Figure 5.1: Fit of the estimated Markov models

one of the states is always 1), whereas a more complex version also considers the loss rate for both states (Gilbert-Elliott model [74]). Figure 5.1 illustrates that the $c_v(n)$ obtained with the latter version closely matches $c_v^{\text{empirical}}(n)$ at all time-scales, whereas the other only matches at short time-scales. The figure also includes the lower bound $c_v(n)$ of a Poisson process with uncorrelated errors.

5.3 Video Quality Evaluation of Impaired Video Sequences

We next turn our attention to the Markov models ability to generate loss patterns yielding visual error patterns in video transmissions similar to the original trace. We therefore impaired video sequences by using three loss traces, *i*) the original trace itself (baseline result), and *ii*) traces generated from the two versions of the Markov chain fitted in the previous section (models). Ideally, visual error patterns resulting from impairing a video sequence with the models (*ii*) closely match the baseline result (*i*). To account for different packet loss rates, we selected different partitions of the original DVB-H packet loss trace and adapted the models to match the mean loss rates for the selected partitions.

We selected five HDTV video sequences, each of 16 seconds length and encoded using H.264 at various bitrates 2, 4, 8, and 16 Mbit/s. The selected video sequences are representative for various kinds of TV content (movie trailer, interview scene, soccer match, movie, music clip) and vary in level of detail and motion complexity, yielding different frame-level properties and encoding efficiency.

Visual properties of impaired video sequences were derived at the frame layer by instructing a video decoder written by Peter List. The modified version of the decoder provided statistical information about decoding errors on a picture level,

e.g., the amount of lost macroblocks or an indicator for freezing. The observed visual impairments depend on the decoder’s error concealment technique, e.g., *freezing* or *slicing*. In the case of freezing, the picture is frozen until the error is completely recovered. Depending on the used decoder version, slicing, however, conceals the error on a macroblock level using motion compensation which may result in visual (block) artifacts within the image.

Focusing on visual error patterns derived from the video decoder allows us to take a *microscopic view* at factors influencing quality. As this work is motivated by supporting quality studies, we aim at optimizing microscopic aspects. Ideally, this optimization should yield model generators that produce similar error patterns as measurement traces. The macroscopic perspective is then added in a later stage by quality studies using the loss generators.

5.3.1 Freezing

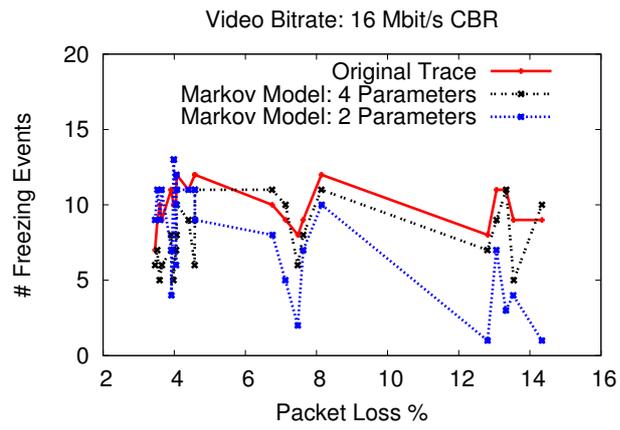
Visual artifacts introduced by *freezing* can be fully characterized by *i)* the amount of freezing events, *ii)* the temporal extent of each event in the number of frozen frames, and *iii)* the total number of frozen frames per video. We show these visual property metrics combined for all five video sequences per loss model as a function of the packet loss rate in Figure 5.2.

The Markov model with two parameters (blue dotted line) departs from the baseline results (trace represented by the red solid line) and shows a high level of variability. In particular, it yields a high variability in the freezing event length distribution for each loss rate (standard deviations not shown in Figure 5.2(b)). In contrast, the more complex Markov model with four parameters (black dashed line) provides a reasonably good fit to the baseline result and yields a lower variability.

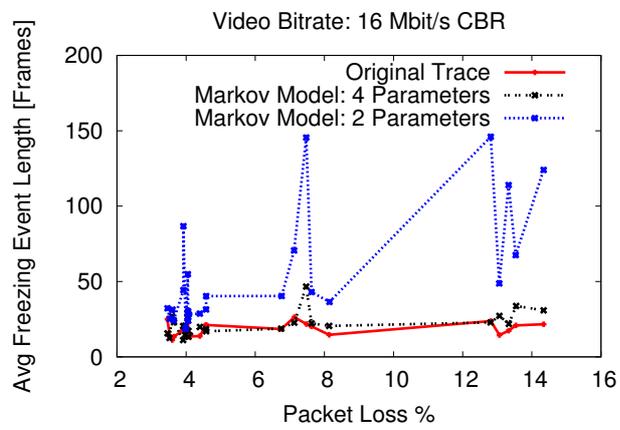
5.3.2 Slicing

Figure 5.3 shows visual properties of the slicing concealment method, i.e., *i)* the total number of frames with lost macroblocks, *ii)* the average number of lost macroblocks per impaired video frame with at least one erroneous macro block, *iii)* the total number of impairment events, and *iv)* the average length of each event. As in Section 5.3.1, each metric is shown as a function of the packet loss rate for each of the three loss sources: the trace representing the baseline result (red solid line), and the two Markov models fitted to the trace.

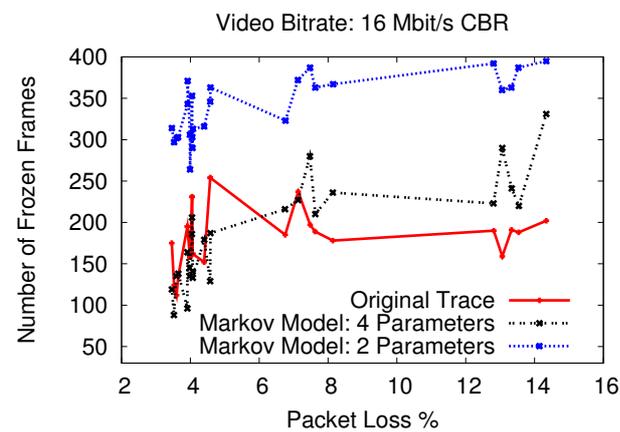
The figures show that compared to the version of the Markov chain with two parameters, the version with four parameters provides a closer match to the considered visual error characteristics, especially at low loss rates. Only the average number of impaired macroblocks is not matched. This suggests that the high variability of video traces require adequate models to generate loss traces with high variability.



(a) # Freezing Events



(b) Freezing Event Length



(c) Total Frozen Frames

Figure 5.2: Visual video quality properties for the freezing concealment method.

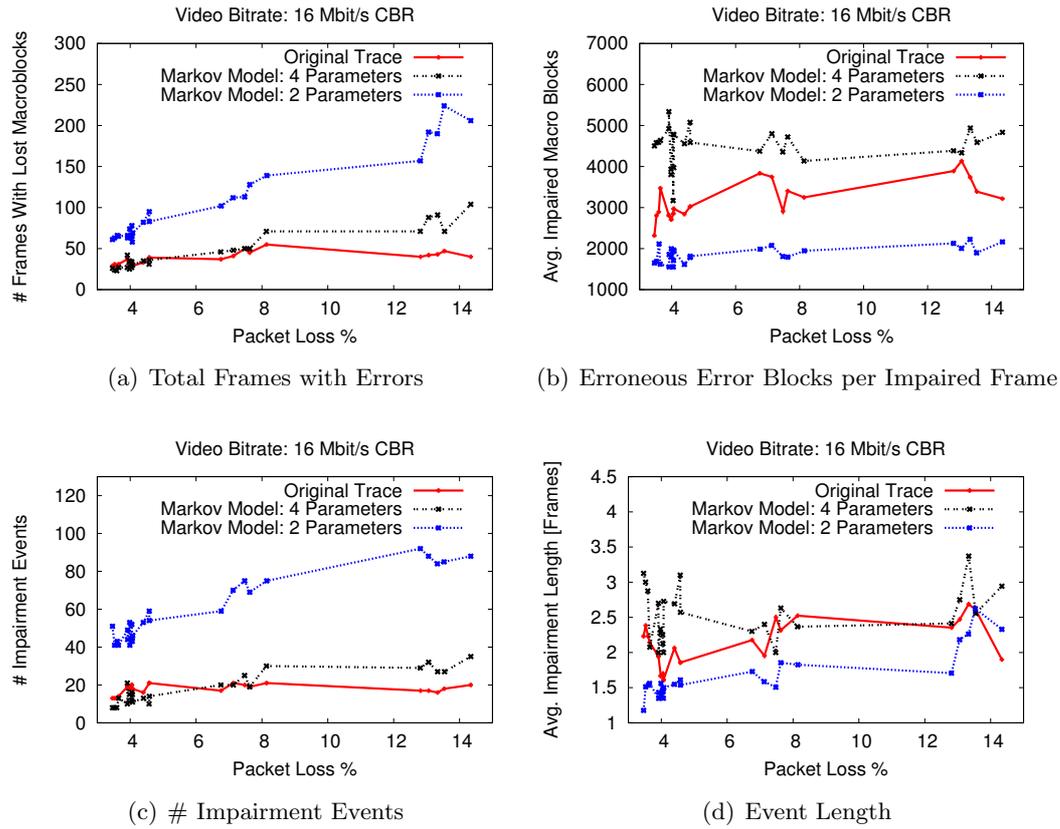


Figure 5.3: Visual video quality properties for the slicing concealment method.

As compared to metrics describing freezing events (cf. Section 5.3.1), metrics describing slicing errors involve much higher variability. This is expressed by large standard deviations (not shown) for metrics shown in Figure 5.3(b) and 5.3(d). Due to overlapping standard deviations, differences in the models are not statistically significant.

5.3.3 Influence of Fitting Method

To highlight the impact of the chosen model and its fitting method, we compare the model summarized in Section 5.2 (second-order statistics) to the following models: *i*) the 2-state Markov chain fitted by applying the Baum-Welch algorithm [247, 231] representing a classical fitting method, and *ii*) a Poisson process generating uniformly distributed packet losses. The impact of the fitting method / model is shown in Figure 5.4 for the average number of freezing events as a function of the packet loss rate. We show this dependence for all the four used video encoding bitrates.

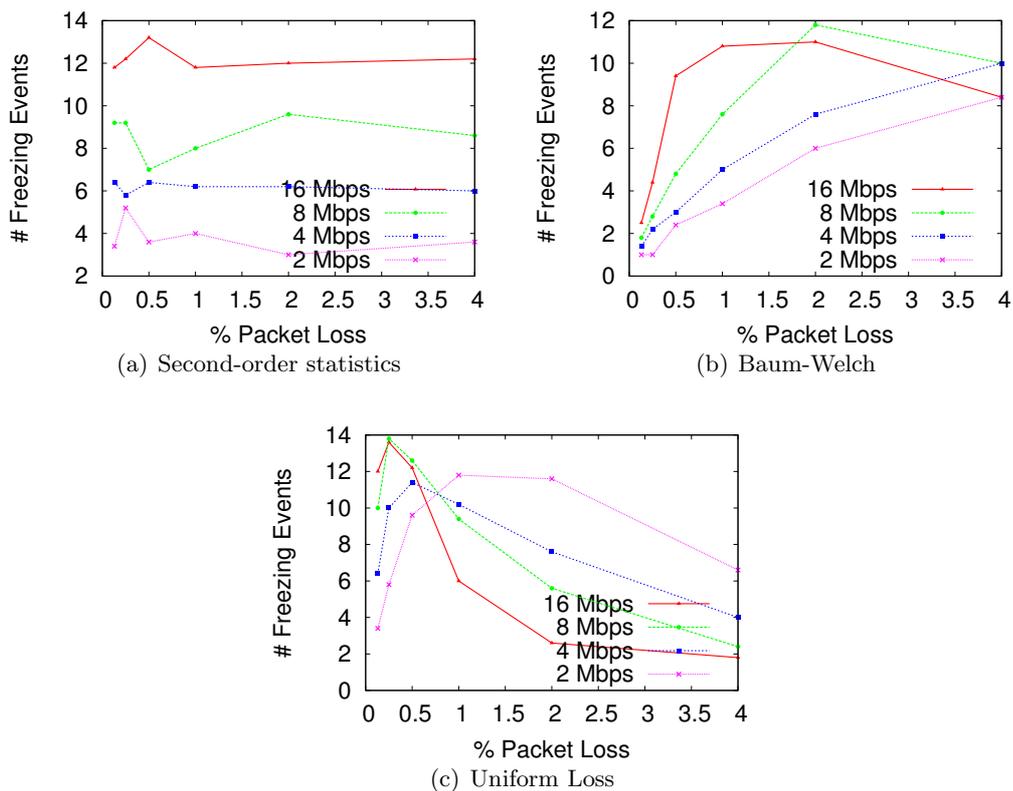


Figure 5.4: Average number of freezing events per video

The obtained results suggest that different modeling approaches do result in different visual error patterns for the same amount of lost packets. This is expressed by different amounts of freezing events per loss rate for each considered fitting technique in Figure 5.4 a), b), and c). As a consequence, it can be said that the choice of the used model and fitting technique therefore matters. The first evaluation using DVB-H traces suggests that a fitting procedure based on second-order statistics (see Section 5.2) more closely replicates visual impairment properties than classical burst length based packet loss modeling techniques (see Section 5.3.1 and 5.3.2).

5.4 Related Work

Markovian channel error models emerged in the 1960s with the classical works of Gilbert [100] and Elliott [74]. These works propose 2-state Markov chains for modelling bit errors in communication channels. Starting in the 1990s, these classical models were applied to packet loss processes that denote a different source of error than bit errors. Example applications include losses in multicast streams over IEEE

802.11 channels [242] and IEEE 802.11 packet error rates [178]. Other works applied these classical works to model loss processes in video transmissions. These applications include datagram losses in UDP video transmissions [131], ATM cell discards in video transmissions [268], DVB video broadcasting [207], and packet losses in Internet video streaming [102]. The use of empirical traces in video quality evaluation and simulation is extensively discussed in [198] from a practical and non-modelling perspective. In contrast to related works, we propose a modelling technique adapted to loss patterns in multiple time scales that is motivated by the structure of video transmissions. Our modelling technique can be applied to classical Gilbert-Elliott 2-state models, as we showed in [117]. In contrast to our previous work that focused on the modelling itself (see [117]), this study evaluated the impact of model choice on quality indicators.

5.5 Discussion

Model based loss generators are used in a variety of studies to examine application performance subject to different packet loss conditions. Examples include network emulation in testbeds, e.g., using the DummyNet network emulator (see for example [88, 56, 60, 155, 160, 111]), or QoE studies (see for example [215]). In contrast to measurement traces that accurately capture the state of the network, model-based loss generators are biased by model limitations. However, they provide the opportunity to precisely control the generated error patterns used in the tests. This is important, as it allows for a mapping between generated error patterns and application performance or QoE.

Motivated by this problem, this chapter discusses a probabilistic packet loss model that is optimized for replicating aspects relevant to video quality. While a large body of work focused on probabilistic packet loss models, they were not designed with QoE aspects in mind. This motivated us to study the impact of different packet loss models on visual error pattern derived from instructing a video decoder. Rather than focusing on QoE directly, we chose to focus on visual error patterns that influence QoE. This choice is motivated by enabling subjective QoE tests using model generators to use stimuli that are as realistic as possible. This permits us to optimize visual error patterns—the stimuli otherwise used in subjective tests—rather than focusing on QoE directly.

In this evaluation we made two observations. First, different packet loss models yield different visual error patterns. This suggests that the loss model choice matters. Second, the approach proposed in Section 5.2 shows good performance in replicating packet loss patterns that yield similar visual error patterns as when impaired with the original trace. In particular, the first-order statistics of the visual error patterns are often matched, while their variability is not sufficiently captured. It may thus

be of interest to consider Markov chain based packet loss generators with a higher state space.

5.6 Future Work

Our study contributed a QoE perspective to packet loss modelling. This represents a first step into a new direction and may lead to more optimized packet loss models, instead of more optimized video quality models developed with non-adequate loss models. Our findings leave us open two directions for future work. The first direction aims at optimizing the choice of time-scales in the fitting process addressing the QoE. Which choice of time-scales results in the most realistic visual error patterns? The second but related direction aims at extensively studying the influence of the state complexity on the models ability to replicate error patterns.

Prior to our study, the QoE community started investigating the QoE of HTTP video streams, e.g., as used by YouTube. Such video streams are based on TCP that retransmits lost packets. Consequently, packet loss translates into delays rather than visual artifacts. Such delays can cause application layer buffer underruns that pause the video playback (freezing). Such freezing conditions are tested by instructing video decoders to pause the playback at preconfigurable times. However, realistic freezing distributions are not yet known. Future work that aims at supporting subjective tests should therefore focus on identifying realistic freezing patterns.

These directions should converge to a broad investigation on the influence of packet loss models on application layer behavior. Such a broad perspective should include multiple empirical measurements and applications. This is particularly relevant as a variety of studies relies on using network emulation (e.g., DummyNet or NetEm) to study application performance subject to packet loss (examples include [88, 56, 60, 155, 160, 111]). As shown in this chapter, loss model choice impacts application layer performance. This opens the question on the appropriateness of different models choices and model parametrizations for different applications. Given that applications can exhibit different behavior when impaired with model generated loss as compared to empirical loss traces, another question concerns the bias introduced by model based loss generators. Answers to these questions are relevant for a broad set of studies relying on loss generators.

6

QoE Impact of Packet Buffers

Packet buffers are widely deployed in network devices to reduce packet loss caused by transient traffic bursts. Surprisingly, even after decades of research and operational experience, ‘proper’ buffer dimensioning remains challenging due to inherent trade-offs in performance metrics applied to the problem. Large buffers can absorb large traffic bursts and therefore increase TCP throughput, but they can also induce significant delays and thus degrade the performance. High delays can result in customer complaints [20] and ultimately lower the revenue of Internet business [226, 170]. Small buffers, on the other hand, can limit end-to-end delays and can reduce costs, but they can also lead to high loss rates that violate service level agreements [233] and cause additional application-layer delays. More broadly, the implications of buffer size choices on application performance are largely unknown from technical, operational, economic, and even perceptual perspectives.

Traditionally, router buffer sizing is proportional to the bandwidth of the linecards, i.e., the bandwidth-delay product (BDP). This rule-of-thumb emerged in the mid 1990s based on studies of the dynamics of TCP flows [136, 250]. A decade later, Appenzeller et al. reexamined buffer sizing and argued that throughput can be maintained using much smaller buffer sizes in core routers [35]. This reignited interest in the research community with regards to buffer dimensioning schemes, however the issue remains far from resolved.

Most recently, the buffer sizing debate has focused on the negative effects of large buffers. The essential argument is that excessive buffering in devices commonly de-

⁰This chapter is joint work with Enric Pujol, Anja Feldmann, and Paul Barford. It has been partially published in [121] in 2012 and is based on a conference submission in 2013.

ployed in the Internet today (aka *bufferbloat*) leads to excessive queuing delays (e.g., in the order of seconds), which negatively influences the performance from a users' perspective [16]. Indeed, *bufferbloat* can adversely effect TCP by increasing round trip times or even triggering unnecessary TCP timeouts. It can also adversely effect UDP by increasing RTTs or packet losses. While such effects have been observed, there is no comprehensive study supporting or disputing the *bufferbloat* arguments. Moreover, despite the absence of empirical evidence, buffer sizes are currently used to drive engineering changes in Internet standards (see e.g., [97]), to re-visit active queue management (AQM) by motivating a new AQM approach (i.e., CoDeL [188]), and to motivate modifications of TCP's congestion control algorithm [105]. None of the currently discussed solutions aims at removing the cause of the problem, i.e., the existence of large buffers, they rather aim at preventing large buffers from being filled. We posit that it is crucial to understand buffering effects before altering important engineering aspects.

In this chapter, we describe the first comprehensive study on the impact of buffer sizes on *end-user perceived QoE* as predicted by state-of-the-art algorithms. The goal of our work is to elucidate the sizing issues empirically and to pave the way for more informed sizing decisions. Unlike previous studies that consider Quality of Service (QoS) metrics (e.g., packet loss events or throughput decay), our study focuses specifically on end-user *Quality of Experience (QoE)*. This is challenging since the users' perception is subjective. However, we argue that end-user QoE is a highly relevant perspective since it has direct implications for network operators and service providers, and by extension, device manufacturers.

To quantitatively evaluate the impact of buffer sizing on estimated end-user experience, we conduct an extensive sensitivity study using real hardware and realistic workload scenarios. Specifically, we evaluate QoE metrics and prediction algorithms for relevant user applications (e.g., web browsing, VoIP, and RTP video streaming) in two realistic laboratory-based testbeds: access and backbone networks. Each application type is analyzed over a range of workloads—without isolation in separate QoS classes—and over a range of buffer sizes.

6.1 Buffering in the Wild

Before investigating the *impact* of buffering on QoE, we first motivate our study by investigating the *occurrence* of buffering in the wild. Our analysis is based on snapshots of Linux kernel level TCP statistics for 430 million randomly selected TCP/HTTP connections captured at a major Content Distribution Network (CDN). The data was collected from different vantage points, located primarily in central Europe, over a period of five months in 2011. All flows were established by end-users to retrieve content from the respective CDN caches, thus they capture typical web browsing activity. This data corpus represents a significant sample of Internet

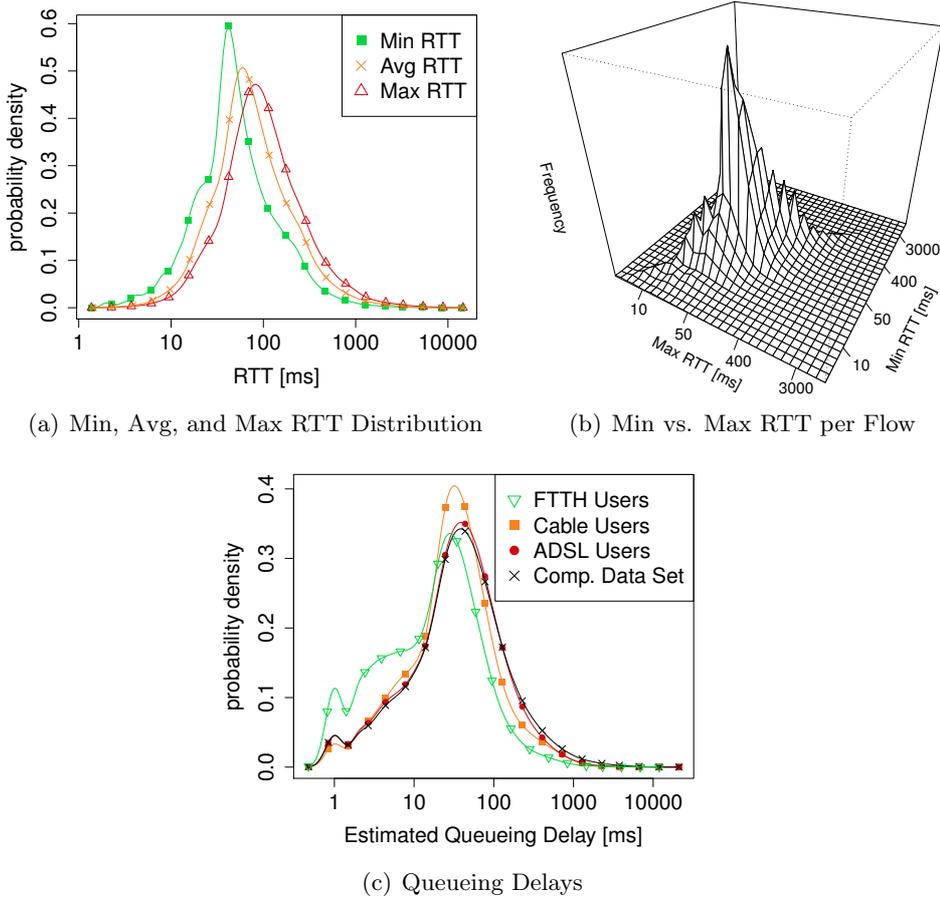


Figure 6.1: Occurrence of queueing in the wild

users. It includes 81 million unique IP addresses originating from 22,490 autonomous systems (roughly 60% of the total advertised ASes in 2011), located in 235 countries. We show the geolocation of the observed IPs in Figure 6.2. Geolocation information is obtained by querying the Maxmind geolocation database.¹ Due to the vantage point locations, 56% of the IPs are located in central Europe.

We build our evaluation on smoothed RTT (sRTT) information reported in the data set. Smoothed RTT values are estimated by the Linux TCP stack using Karns algorithm [146] and are provided by the kernel level TCP statistics. Specifically, for each TCP connection, the data set reports *(i)* the minimum sRTT, *(ii)* the average sRTT, *(iii)* the maximum sRTT, and *(iv)* the number of samples. To evaluate the variability due to queueing, we focus on flows that have at least 10 RTT samples. The

¹We remark that geolocation databases such as Maxmind have been shown to be modestly accurate on a city-level [78, 206], but fairly accurate on a country-level [206]. For the purpose of illustrating the broad geographic spread of users in the considered data set, country-level accuracy, however, is sufficient.

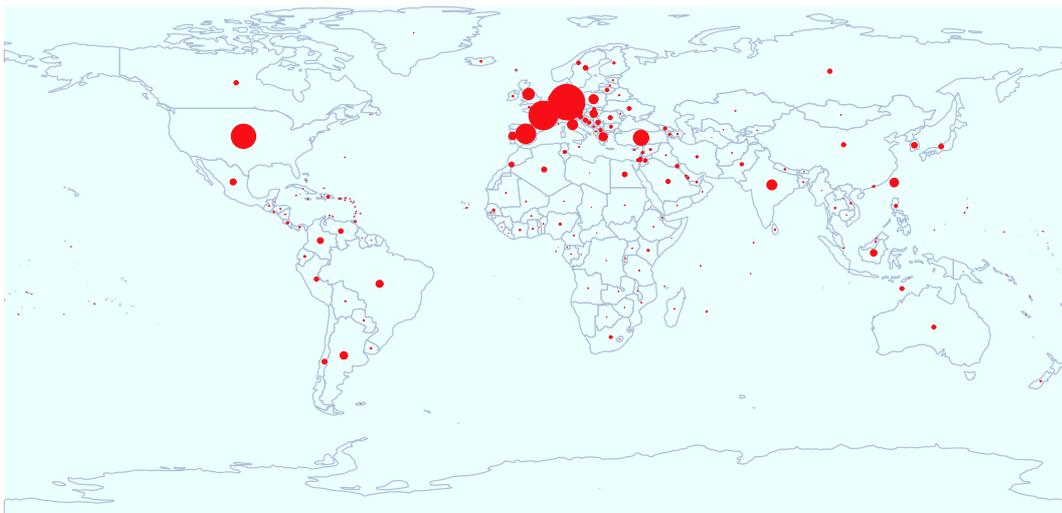


Figure 6.2: End-user geolocations

distribution (PDF) of the logarithm of the minimum, average, and maximum RTT is shown in Figure 1a. The plot highlights that the average and maximum RTT deviate significantly from the minimum RTT, which is one indicator of possible queueing. Figure 1b underlines this intuition by showing the relationship of minimum and maximum RTT per flow in a 2D histogram. The figure shows that the maximum RTT significantly differs from the minimum RTT per flow, which further suggests the presence of queueing.

We estimate the queueing delay by evaluating the sRTT range (i.e., max-min) for each connection with at least 10 RTT samples. The implicit assumption is that the minimum RTT accounts for an empty queue and that queueing is the only source of delay variations. In general, additional factors such as route changes and layer 2 delays—particularly prominent in wireless networks—also contribute to delay variations. Since we cannot separate these factors from queueing delays, our estimation conceivably overestimates queueing and thus yields an *upper bound* on the magnitude of queueing.

We show the PDF of the logarithm of the estimated queueing delay in Figure 1c. Based on whois and DNS information, we split the complete data set into ADSL, Cable, and FTTH users and show their respective queueing delay distribution. Using this scheme, we associate 70% of the flows to ADSL users, 1.4% to Cable users, and 0.02% to FTTH users. Most of the user flows experience a modest amount of queueing; 80% of all the flows experience less than 100ms of delay variation. Only 2.8% (1%) of the observed flows experience excessive queueing delays of more than 500ms (1000ms). This corresponds to only 2.5% (2%) of the observed hosts. We also consider user proximity to the CDN caches. Specifically, we consider flows

with minimum RTT ≤ 100 ms. In this setting, even more flows experience modest amounts of queueing: 95% (99.9%) of all connections have a queuing delay of less than 100ms (1sec), respectively.

Recently, the issue of *buffer bloat* has attracted significant attention. The debate is based on observations (e.g., [158]) showing that bufferbloat *can* happen, rather than it *does* happen. Despite this lack of empirical evidence, the bufferbloat argument has been used to motivate engineering changes in Internet standards (e.g., see [97]) and to motivate new AQM approaches (e.g., CoDeL [188]). Two very recent studies examined the magnitude of the problem based on data from 118K [33] and 25K hosts [59], respectively and concluded that the magnitude of bufferbloat is modest.

Our results above, based on a much larger data set of 80M hosts that is representative for a significant body of Internet users, further substantiate these findings. We empirically study whether Internet users at large experience excessive delays and we conclude that excessive delays do occur, but only for a small number of flows and hosts. Thus—despite what is often claimed by the bufferbloat community—our findings further confirm the modest magnitude of the problem. One explanation is that uplink capacity in the access, where bufferbloat has been found, is seldom utilized.

Our study of buffering in the wild is the starting point for our evaluation of the impact of buffering on QoE, including the case of excessive buffering (bufferbloat). While we estimate the magnitude of bufferbloat to be modest, its implications on QoE are largely unknown. For instance, a single delayed flow can severely degrade the QoE of an entire HTTP transaction. To shed light on the QoE impact of buffering, we conduct an extensive study covering a wide range of end-user applications, buffer configurations, and traffic scenarios.

6.2 Measurement Setup

We use a testbed-driven approach to study the impact of buffer sizes on the user perceived QoE of common types of Internet applications: *i*) Voice over IP, *ii*) RTP/UDP video streaming as used in IPTV networks, and *iii*) web browsing, using QoE estimates derived from respective algorithms.

6.2.1 Testbed Setup

We consider two scenarios: *i*) an access network and *ii*) a backbone or core network. Each scenario is realized in a dedicated testbed as shown in Figure 6.3 (a) and (b). We use a testbed setup to have full control over all parameters including buffer sizes and generated workload.

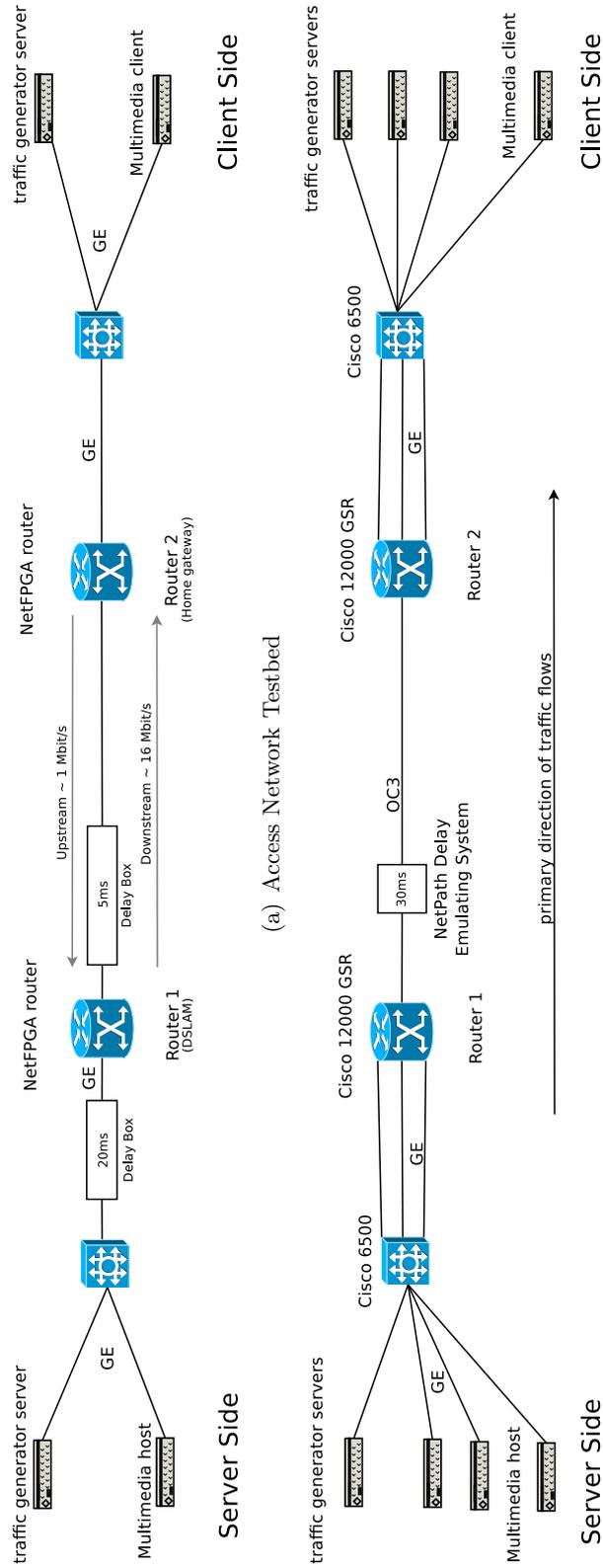
As most flows typically experience only a single bottleneck link, both testbeds are organized as a dumbbell topology with a single bottleneck link, configurable buffer sizes, and a client and a server network. The hosts within the server (client) network on the left (right) side act as servers (clients), respectively. In the backbone case, we configured the bandwidth and the delays of all links symmetrically. For the access network, we use an asymmetric bottleneck link. In the backbone case, we only consider data transfers from the servers to the clients. For the access network we also include data uploads by the clients—as they mainly triggered the bufferbloat debate [98].

The access network testbed, see Figure 6.3(a), consists of two Gigabit switches, four quadcore hosts equipped with 4 GB of RAM and multiple Gigabit Ethernet interfaces. Moreover, two hosts are equipped with a NetFPGA 1 Gbit/s card each. The hosts are connected via their internal NICs to the switch to realize the client/server side network. The NetFPGA cards run the Stanford Reference Router software and are thus used to realize the bottleneck link. The NetFPGA router and the multimedia hosts are located on the same physical host. As the NetFPGA card is able to operate independent of the host, it does not impose resource contention. The right NetFPGA router acts as the home router, aka DSL modem, whereas the left one acts as the DSLAM counterpart of the DSL access networks. To capture asymmetric bandwidth of DSL, we use the hardware capabilities of the NetFPGA card to restrict the uplink and downlink capacities to approximately 1 respectively 16 Mbit/s. We use hardware to introduce a 5 ms respectively 20 ms delay between the client (server) network and the routers. The 5 ms delay corresponds to DSL with 16 frame interleaving or to the delays typical for cable access networks [41]. The 20 ms account for access and backbone delays. While we acknowledge that delays to different servers vary according to a network path, a detailed study of path delay variation is beyond the scope of this study. This is also why we decided to omit WiFi connectivity, which adds its own variable delay characteristics due to layer-2 retransmissions. Instead, we focus on delay variations induced by buffering.

To be able to scale up the background traffic to the backbone network, see Figure 6.3(b), we include eight hosts, i.e., four clients and four servers. Each has again a quadcore CPU, 4 GB of RAM, and multiple Gigabit Ethernet network interfaces. The client/server networks are connected via separate Gigabit switches, Cisco 6500s, to backbone grade Cisco 12000GSR routers. Instead of using 10 Gbit/s and soon to be 100 Gbit/s interfaces for the bottleneck link, we use an OC3 (155 Mb/s nominal) link. The reason for this is that we wanted to keep the scale of the experiments reasonable, this includes, e.g., the tcpdump files of traffic captures. Moreover, without limiting the validity of this study, scaling down allows us to actually experience bufferbloat given the available memory within the router. We use multiple parallel links between the hosts, the switch, and the router so that it is possible for multiple packets to simultaneously arrive at the router buffer. With regards to the delays, we added a NetPath delay box with a constant one-way delay of 30 ms to the bottleneck link. 30 ms delay roughly corresponds to the one-way delay from the US east

to the US west coast. We again note, that the path delays in the Internet are not constant. However, variable path delays are beyond the scope of this study. Instead we focus on delay variability induced by buffering. Moreover, we eliminate most synchronization potential by our choice of workload (see Section 6.2.2).

To gather statistics and to control the experiments we always use a separate Ethernet interface on the hosts and separate physical network (not shown).



(b) Backbone Network Testbed

Figure 6.3: Testbeds used in the study

| Testbed | Scenario Name | Flow Interarrival Distribution | File Size Distribution | # Sessions | Concurrent Flows | Link Utilization [%] | | Packet Loss [%] | | Description | | | |
|----------------|---------------|--------------------------------|------------------------|------------|------------------|----------------------|------|-----------------|------|-------------|------|----------------------------|----------------|
| | | | | | | Mean | Sd | Up | Down | | Up | Down | |
| Access | noBG | — | — | — | — | 98.9 | 0.3 | 0.7 | 0.1 | 34.7 | 0 | No bg. traffic Upstream | |
| | short-few | exp-a | weibull | 1 | 8 | 95 | 8.5 | 5.6 | 15.2 | 58.6 | 0.7 | Bidirectional | |
| | | | | 1 | 8 | 27.8 | 44.1 | 13.7 | 25.1 | 1.4 | 3 | Downstream | |
| | short-many | exp-a | weibull | 1 | 16 | 98.9 | 0.3 | 0.7 | 0.1 | 33.1 | 0 | Upstream | |
| | | | | 1 | 16 | 93.3 | 10.7 | 4.3 | 20.1 | 60.9 | 1.3 | Bidirectional | |
| | long-few | — | infinite | 1 | 8 | 53.8 | 78.7 | 12.8 | 23.5 | 4 | 4.5 | Downstream | |
| | | | | 1 | 8 | 99 | 0.2 | 0.7 | 0.0 | 1 | 0 | Upstream | |
| | long-many | — | infinite | 1 | 8 | 71.9 | 83.1 | 8.9 | 12.6 | 41.7 | 0.6 | Bidirectional | |
| | | | | 8 | 64 | 39.5 | 99.9 | 1.9 | 0.6 | 0.1 | 0.5 | Downstream | |
| | Backbone | noBG | exp-b | weibull | 8 | 64 | 98.9 | 0.3 | 0.7 | 0.0 | 14.4 | 0.0 | Upstream |
| 8 | | | | | 64 | 83.8 | 61.8 | 11.2 | 26.4 | 60.7 | 0.2 | Bidirectional | |
| Backbone | short-low | exp-b | weibull | 8 | 64 | 68.5 | 99.6 | 3.9 | 4.9 | 0.03 | 9.3 | 0 | Downstream |
| | | | | 3 * 10 | 18 | 16.5 | — | 11.6 | — | — | — | 0 | No bg. traffic |
| | | | | 3 * 30 | 49 | 49.5 | — | 18.8 | — | — | — | 0 | — |
| | | | | 3 * 60 | 206 | 98 | — | 6.5 | — | — | — | 0.2 | — |
| short-overload | exp-b | weibull | 3 * 256 | 2170 | 99.7 | — | 2.2 | — | 5.2 | — | — | | |
| | | | 3 * 256 | 675 | 99.7 | — | 0.1 | — | — | 3.8 | — | | |

Table 6.1: Workload configurations for both testbeds, where the flow interarrival time distributions are specific to the access and backbone testbed; exp-a has a mean of 2 sec and exp-b a mean of 1 sec. The “weibull” file size distribution is defined as Weibull (shape=0.35, scale=10039), resulting in a mean flow size of 50 KB. The number of concurrent flows at the bottleneck link is shown by their mean. Link utilization and loss measures are obtained for buffers configured according to the BDP. See text for further details.

6.2.2 Traffic Scenarios

We use the Harpoon flow level network traffic generator (see Section 2.2.3) to create a number of congestion scenarios which range from no background traffic (**noBG**) to fully overloading (**short-overload**) the bottleneck link. Congestion causes packets from both the background traffic as well as the application under study to be queued or dropped just before the bottleneck link. Depending on the fill grade of the buffer, the size of the buffer, and the link speed, this will increase the RTT accordingly (see Table 6.2). Overall, we use twelve scenarios for the access testbed and six for the backbone. We consider more for the access as we distinguish on which links, upstream, downstream, or both, the congestion occurs.

In terms of traffic that imposes the congestion, we distinguish two different kinds of scenarios (see Table 6.1): (i) long-lived TCP flows (**long**) and (ii) long-tailed file sizes to be able to resemble self-similar traffic as seen in today's networks. For the latter, we choose Weibull distributed file sizes with a shape of 0.35 as their mean and standard deviation is finite, as opposed to those of the often used Pareto distributions with a shape > 2 . The generated traffic results in a mixture of bursty short-lived and long-lived flows with a mean of 50 KB. As the number of short flows dominates the number of long flows, we refer to these scenarios as “**short**”.

For scenarios with long-lived flows (**long**), we use flows of infinite duration. In this case, the link utilization is almost independent of the number of concurrent flows. For long-tailed file sizes, the workload of each scenario is controlled via the number of concurrent sessions that Harpoon generates. A session in Harpoon is supposed to mimic the behaviour of a user [234] with a specific interarrival time, a file size distribution, and other parameters (see Section 2.2.3). We used the default parameters except for the file size distribution. In addition, we rescaled the mean of the interarrival time for the access network, as Harpoon's default parameters are geared towards core networks with a larger number of concurrent flows. To impose different levels of congestion, we adjusted the number of sessions for the backbone scenario to result in *low*, *medium*, *high*, and *overload* scenarios which correspond to link utilizations as shown in Table 6.1. For the access network we distinguish between few and many concurrent flows which results in medium and high load for the downstream direction and high load for the upstream, see Table 6.1.

We checked that all hosts are using a TCP variant with window scaling. Due to the Linux version used, the background traffic uses TCP-Reno in the backbone and TCP BIC/TCP CUBIC for the access. However, note that this does not substantially impact the QoE results, as the applications VoIP and video rely on UDP and the Web page is relatively small. Moreover, since the results are consistent it highlights that using a TCP variant optimized for high latency does not change the overall behavior, even if the buffers are large.

| Buffer Size (Pkts) | Access | | | | Backbone | | |
|--------------------|------------|---------------|------------|---------------|--------------------|------------|-------------------|
| | Uplink | | Downlink | | Buffer Size (Pkts) | Delay (ms) | Scheme |
| | Delay (ms) | Scheme | Delay (ms) | Scheme | | | |
| 8 | 98 | \approx BDP | 6 | min | 8 | 0.6 | \approx TinyBuf |
| 16 | 198 | | 12 | | 28 | 2.2 | Stanford |
| 32 | 395 | | 24 | | 749 | 58 | BDP |
| 64 | 788 | | 49 | \approx BDP | 7490 | 580 | $10 \times$ BDP |
| 128 | 1,583 | | 97 | | | | |
| 256 | 3,167 | max | 195 | max | | | |

Table 6.2: Buffer size configurations and corresponding maximum queuing delays for both testbeds (packet size = 1500 bytes)

6.2.3 Buffer Configurations

One key element of our QoE study are the buffer size configurations. Buffers are everywhere along the network path including at the end-hosts, the routers, and the switches (see Section 2.4). The most critical one is at the bottleneck interface, the only location where packet loss occurs. Therefore we focus on these and rely on default parameters for the others. For the bottleneck we choose a range of different buffer sizes, some reflect existing sizing recommendations, some are chosen to be small, others large in order to capture extremes. Table 6.2 summarizes the buffer size configurations in terms of number of packets and shows the corresponding queuing delays.

For the access network, we choose buffer sizes of powers of two, ranging from 8 to 256 packets. 256 is the maximum supported buffer size by the Stanford Reference Router software. Further, we assume symmetric buffer size configurations for both the uplink and downlink direction and leave the asymmetric case for future work. For our choice of an asymmetric link (recall 1 Mbps uplink/16 Mbps downlink) the bandwidth-delay product (BDP) corresponds to roughly 8 and 64 packets, respectively.

For the backbone network we use *i)* the same minimum buffer size of 8 packets, which resembles the TinyBuffer scheme [76], depending on the largest congestion window achieved by the workloads. In addition, we use *ii)* 749 full-sized packets which corresponds to the BDP formula given an RTT of 60 ms, *iii)* 28 packets which corresponds to the Stanford scheme [35], i.e., BDP/\sqrt{n} , where $n = 3 * 256$ is the maximum number of concurrent flows for short-low, short-medium, short-high, and long (see Table 6.1), and *iv)* $10 \times$ BDP packets as an excessive buffering scheme.

6.3 QoS: Background Traffic

To highlight the potential importance of the buffer configuration on latencies, network utilization, and packet loss—the typical QoS values—we start our study with a detailed look at the background traffic. While the story is relatively straight forward for the backbone scenario, and captured in Table 6.1, it is more complicated for the access network, as the number of concurrent flows is smaller and there are subtle interactions between upstream and downstream.

To illustrate how the workloads and buffer sizes impact real-time applications, we conduct experiments to measure the latency introduced by the buffers. For this purpose, we use the detailed buffer-fill statistics of the FPGA cards. Figure 6.4 shows the corresponding mean delays as heatmaps. We use three different heatmaps: one each for downstream/upstream workload only and one for combined up- and downstream workload. Each heatmap has two subareas—one for upstream at the top and one for downstream at the bottom. The values in the heatmaps show the mean delays based on a specific buffer size configuration and a specific workload scenario. Each value is based on one single experiment of two hours length including a multitude of delay measurements. The color of the heatmap cells are chosen according to ITU-T Recommendation G.114, which classifies delays based on their potential to degrade the QoE of interactive applications: green (light gray) is acceptable, orange (medium gray) problematic, and red (dark gray) causes problems (for a detailed classification of delay based speech quality degradations as estimated by the ITU E-Model, see ITU G.114).

In principle, we see that larger buffer sizes can increase the delays significantly, independent of the workload. For the downlink direction, the maximum delay is less than 200 ms. However, this can differ for the uplink direction. In particular, we observe delays of up to three seconds for larger, over-sized-buffers when the upstream is utilized. Overall, these delays are consistent with observations by Gettys [16] which started the bufferbloat discussion.

Given these high latencies, we investigate the link utilization. Figure 6.5 shows a boxplot of the link utilization for the specific scenario with simultaneous downloads and uploads (bidirectional workloads) for the various buffer sizes. The left/right half focuses on the downlink/uplink utilization. We see that the uplink utilization is almost 100% while the downlink utilization ranges from 20% to 100%. Thus, we see that very small buffers can lead to underutilization while very large buffers can lead to large delays.

Comparing these link utilizations to those with no upstream workload (not shown) we find that, for bidirectional workloads, the buffer configurations below the BDP do not always fully utilize the downlink direction. Buffer sizes that correspond to the BDP yield full downlink utilization in the absence of upload workload, but not with concurrent download and upload activities. These phenomena can be explained by

the queuing delay introduced by bloated uplink buffers that *virtually* increase the BDP, thus rendering the downlink under-buffered.

The phenomena of low link utilization can be mitigated by counter-intuitively “bloating” the downlink buffer. Considering the delays observed in Figure 6.4(c), the BDP increases beyond the initial buffer size of 64 to 835 full sized packets. Note, that we can get full link utilization for buffers of smaller than 835 packets as we have a sufficient number of concurrent flows active.

In summary, the latency introduced by the buffers in home routers, aka, the uplink, might not only i) harm real-time traffic applications (due to excessive buffering), but also ii) drastically reduce TCP performance (due to insufficient buffering) in case of bidirectional workloads in asymmetric links. In effect it invalidates the buffer dimensioning assumptions due to the increase in RTT.

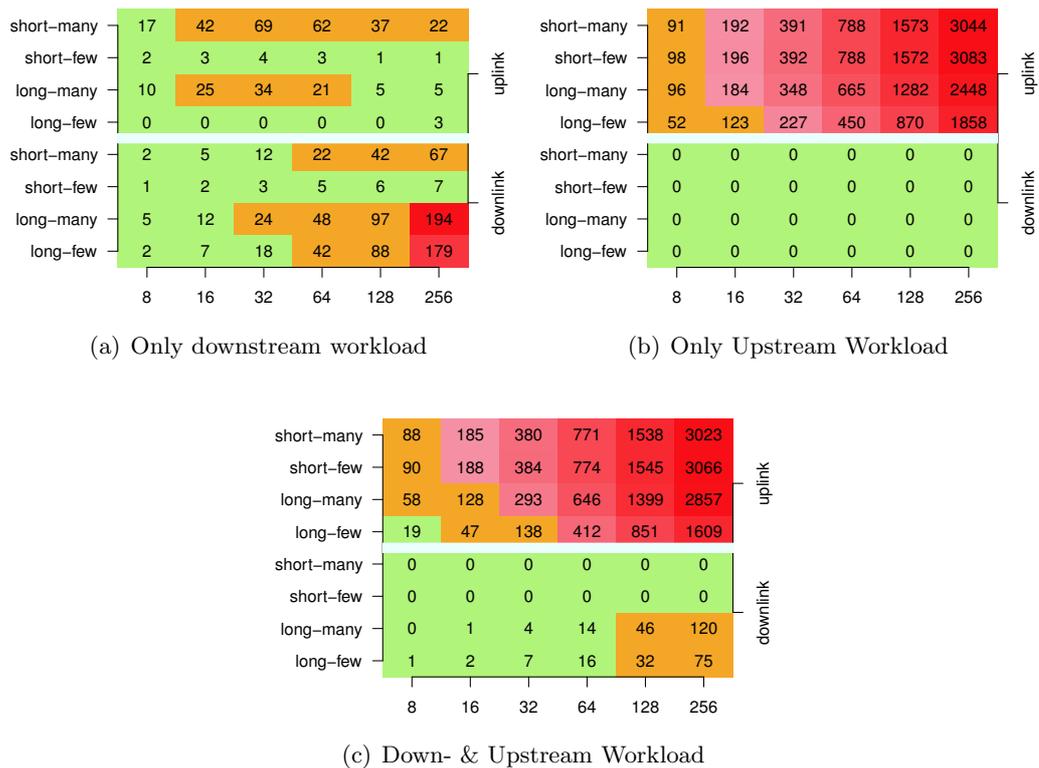


Figure 6.4: Mean queuing delay (ms) for the access networks. Delays likely to degrade QoE colored in red.

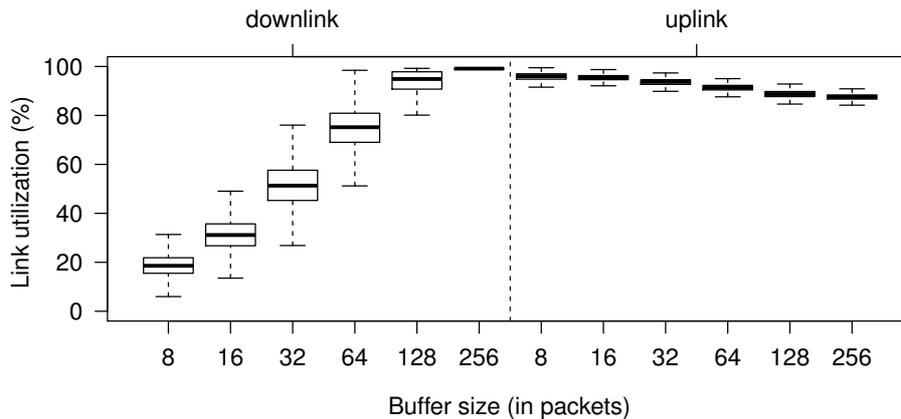


Figure 6.5: Link utilization in an asymmetric access link

6.4 Voice over IP

We start our discussion of application QoE with Voice over IP (VoIP). In IP networks, speech signals can be impaired by packet loss, jitter, and/or delay. To be more specific, packet losses directly degrade speech quality, as long as forward error correction is not used. Network jitter can result in losses at the application layer, as the data arrives after its scheduled playout time. This equally impacts speech quality. Moreover, excessive delays impair any bidirectional conversation as it changes the conversational dynamics in turn taking behavior (see Section 3.3.1).

6.4.1 Approach

We use a set of 20 speech samples recommended by the ITU [12] for speech quality assessment. Each sample is an error-free recording of a male or female Dutch speaker, encoded with G.711.a (PCMA) narrow-band audio codec, and lasts for eight seconds.

Each of the 20 samples is automatically streamed, using the PjSIP library, over our two evaluation testbeds, see Section 6.2 and subjected to the various workloads. PjSIP uses the typical protocol combination of SIP and RTP for VoIP. We remark that we do not consider other situational factors such as the users' expectation (e.g., free vs. paid call) [183] which can also affect the perceived speech quality (see Chapter 3).

For evaluating the QoE of the voice calls we first evaluate the loss and jitter effects, and then the delay effects, using two QoE models, PESQ and the E-Model; the resulting scores are then combined to the final QoE score (see Section 3.3.1).

Loss and Jitter effects. To assess the speech quality of each received output audio signal, relative to the error-free sample signal, we use the Perceptual Speech Quality Measure (PESQ) [9] model. PESQ takes as input both the error-free signal and the perturbed signal, and computes the estimated QoE score z_1 .

Delay effects. Speech quality is impacted by two types of delay: *i)* pure delay impacts call interactivity and *ii)* talker echo. As our focus remains on networking delay, we assume well-working echo canceler so that talker echo does not lead to quality degradations in this study.

The PESQ model only accounts for the perceived quality when listening to a remote speaker, but does not account for conversational dynamics, e.g., for humans taking turns and/or interrupting each other. This can be impaired by excessive delays and thus can degrade the quality of the conversation significantly [183, 150, 212, 222, 72, 113]. Thus, according to the ITU-T recommendation G.114 one-way delays should be below 150 ms (or at most 400 ms).

Therefore, we measure the packet delay during the VoIP calls. We now use the delay impairment factor of the ITU-T E-Model [17] to get a score z_2 . We remark that even though z_2 is computed using a standardized and widely used model, it is subject to an intense debate within the QoE literature, since there is a dispute about the impact of delay on speech perception [150, 212, 73, 232]. Among the reasons is that the delay impact depends on the nature of the conversational task (e.g, reading random numbers vs. free conversation) as well as the level of interactivity required by the task [150]. Thus, there can be mismatches between the quality ratings of the E-Model and tests conducted with subjects. We remark that respective updates of the E-Model have been proposed in [232].

Overall score. The range of the score z_1 , which captures loss and jitter, is [1, 5]. We remap it to [0, 100] according to [239]. The range of the score z_2 , capturing the delay impairment, is [0, 100]. Note, the semantics of z_1 and z_2 are reversed: a large value for z_1 reflects an excellent quality; however, a large value for z_2 reflects a bad quality, and vice-versa. We combine the two scores to an overall one as follows: $z = \max\{0, z_1 - z_2\}$. Thus, if z_1 is good (i.e., due to negligible loss and jitter), but the z_2 is bad (i.e., due to large delays), then the overall score z is low, reflecting a poor quality and vice-versa. Finally, we map z to the MOS scale [1, 5] according to the ITU-R recommendation P.862.2, see Figure 6.6(a); in the end, low values correspond to bad quality and high values to excellent quality.

6.4.2 Access Networks

Figures 6.7(a) and 6.7(b) show heatmaps of the median call quality (MOS) for the access networks. Each cell in the heatmap shows the median MOS of 200 VoIP calls (each speech sample is sent 10 times) per buffer size (x-axis) and workload scenario (y-axis) combination. The heatmap is colored according to the color scheme of

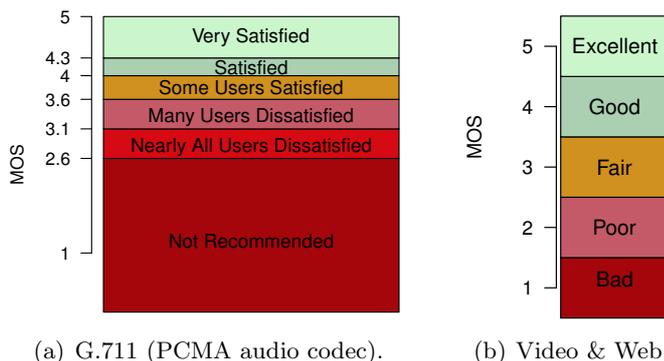


Figure 6.6: MOS scales used in this study

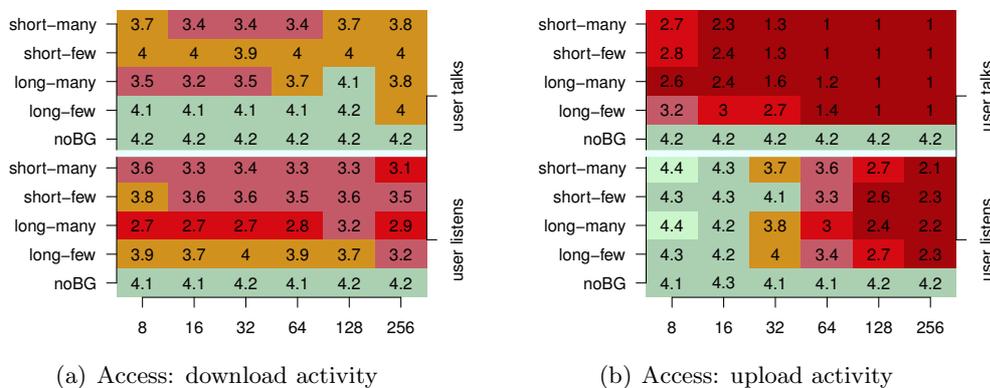


Figure 6.7: Median Mean Opinion Scores (MOS) for voice calls with different buffer sizes and workloads

Figure 6.6(a). The heatmap is divided into two parts: (i) when user talks (upper part) and (ii) when the user listens to the remote speaker (bottom part).

The baseline results, namely the ones without background traffic are shown in the bottom row of each heatmap part, labeled `noBG`. They reflect the achievable call quality of the scenarios. As all of them are green, we can conclude that in principle each scenario supports excellent speech quality and that any impairment is due to congestion and not due to the buffer size configuration per se.

Download activity. Figure 6.7(a) focuses on the scenarios when there is congestion in the downlink. As there is no explicit workload in the uplink, one may expect that only the “user listens” part is affected but not the “user talks” part. This is only partially true as the “user talks” part of the heatmap shows deviations of up to 0.8 MOS points from the baseline score. These degradations are explained by the

substantial number of TCP ACK packets, reflected by higher link utilizations (not shown). Recall, the uplink capacity is only 1/16th of the downlink capacity.

The degradations in the “user listens” part of the heatmap are, as expected, more pronounced than for the “user talks” part. However, there are also significant differences according to the workload and the buffer configurations. For instance, with buffers sizes of 64 packets the long-many workload yields a median MOS of 2.8, whereas the long-few workload yields a median MOS of 3.5. Interestingly, even though the short-few workload does not fully utilize the downlink, i.e., less than 50% (not shown), it gets scores worse than a workload with higher link utilization, e.g., long-few. This is due to the higher jitter that is imposed by the large changes in link utilization and thus in the buffer utilization. With regards to buffer sizes we observe the worst scores for the larger buffer configurations, i.e., 256 packets, due to the added delays. However, the best scores only deviate by 0.7 MOS points from this worst score (e.g., for the 8 packets buffer), suggesting that smaller buffers do not significantly improve audio quality.

We conclude that the level and kind of workload has a more significant effect than buffer size.

Upload activity. Figure 6.7(b) focuses on the scenarios when there is congestion in the uplink. According to the above reasoning one would thus only expect degradations for the “user talks” part. This is not the case. The MOS in the “user listens” part of the heatmap decreases by 0.5 to 2 from the baseline results for all scenarios with buffer sizes ≥ 64 . The reason for this is that the delays added by the excessive buffering in the uplink also degrade the overall score due to the delay impairment factor z_2 . Since this factor expresses the conversational quality, it does not only effect the “user talks” but also the “user listen” part sent over the (non-congested) downlink.

The excessive queuing added by the buffers also explain the significant degradation of MOS values from 4.2 to 1–1.4 for the “user talks” part. But due to the congestion, packet loss is also significant for all scenarios. This is the reason why the best MOS value is limited to 3.2 even for short buffer configurations.

In the context of the bufferbloat discussion, Figure 6.7(b) corroborates that excessive buffering in the uplink indeed yields bad quality scores. Reducing the buffer sizes results in better MOS and contributes to mitigate the negative effects of the large delays introduced by the uplink buffer, e.g., the difference in the MOS for an inbound audio can be as high as 2.5 points (long-many workload).

Combined upload and download activity. Scenarios (plot not shown) with upload and download congestion show similar results to scenarios with only upload traffic. Here as well the delays introduced by the uplink buffer dominate in both “user talks” and “user listens” parts. However, with combined upload and download activity, the “user listens” is slightly more degraded than with only upload activity. The reason for this is additional background traffic in the downlink that interacts

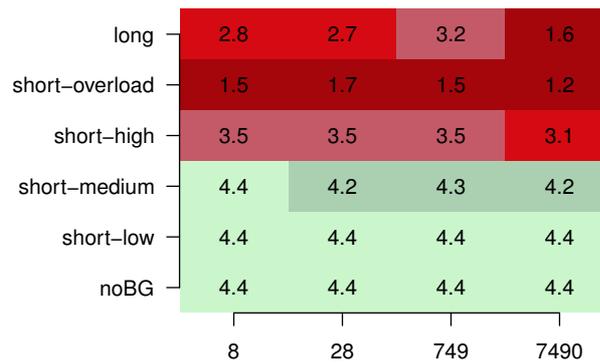


Figure 6.8: Median VoIP MOS scores for backbone networks

with the voice call. For instance, with buffers configured to 16 packets, the *long-few* scenario leads to an additional degradation of 0.8 MOS points, thus leading to an even lower satisfaction level (see Figure 6.7).

Based on these findings, it is generally a good strategy to isolate VoIP calls in a separate QoS class when the access link is subject to congestion.

6.4.3 Backbone Networks

Similar to the access network scenario, we show the voice quality in the backbone network scenario as a heatmap in Figure 6.8. The heatmap shows the median MOS for unidirectional audio from the left to the right side of the topology per buffer size (x-axis) and workload scenario (y-axis). Each cell in the heatmap is based on 2000 VoIP calls. Here, each speech sample is sent 100 times which is possible as the total number of scenarios is smaller. As in the access network scenario, the bottom row label *noBG* shows the baseline results for an idle backbone without background traffic.

While the effects of the buffer size are less pronounced, impacts due to the nature of the background traffic (*long* vs. *short-**) and the link utilization (*short-low* to *short-overload*) are more significant. The type of workload can drastically degrade the quality score. Concretely, low to medium utilization levels as imposed by the scenarios *short-low* and *short-medium*, respectively, lead to quality estimates close to the baseline results. In contrast, more demanding workloads such as the scenarios *short-high* and *long*, leading to higher link utilizations, and result in more than 1 point reductions in the terms of MOS. Further, the aggressiveness of the workload further decreases the quality; the median MOS for the *short-overload* workload is 1.5 and thus significantly lower than for *short-high* and *long* that also lead to high link utilizations.

In general, the quality scores are, on a per workload basis, fairly stable across buffer-configurations below the BDP (749 packets). In these cases, we only observe a small degradation of 0.4 points for the scenario long workload for the smallest buffer configurations. However, buffer configuration larger than the BDP, i.e., 7490 packets, lead to excessive queuing delays. As in the access network scenario, excessive delays lead to significant quality degradations of the z_2 delay impairment component. For example, the scores corresponding to the scenarios long and short-overload workloads lead to MOS values of almost half of their counterpart with the BDP configuration.

6.4.4 Key Findings for VoIP QoE

We find that VoIP QoE is substantially degraded when VoIP flows have to compete for resources in congested links. This is highlighted in the backbone network scenario, where low to medium link utilizations yields good QoE and high link utilization ($> 98\%$) degrade the QoE. In the case of the latter, the congestion leads to insufficient bandwidth on the bottleneck link that affects the VoIP QoE.

For access networks we show that, due to the asymmetric link capacities, the different audio directions can show different QoE scores. For instance, in one direction (e.g., user talks) the voice might be relatively small or not affected, while it is impaired for the other (e.g., remote speaker talks) or vice-versa. Moreover, the speech quality is much more sensitive to congestion on the upstream direction than the downstream one. Due to the light queuing delays introduced by bloated buffers in the uplink, maintaining an interactive conversation can be challenging in the presence of uplink congestion.

For both access and backbone networks, configuring small buffers can result in better QoE. However, our results highlight that this may not suffice to yield “excellent” quality ratings. Thus, we advocate to use QoS mechanisms to isolate VoIP traffic from the other traffic. We remark that some ISPs already use QoS for improving the service for their managed services (e.g., ISP-internal IPTV). This, however, typically is not the case for over-the-top applications (e.g., Skype or third-party VoIP services).

6.5 RTP Video Streaming

We next explore the quality of video streaming using the Real-time Transport Protocol (RTP), which is commonly used by IPTV service providers. RTP streaming can be impaired by packet loss, jitter, and/or delay. Again, packet losses directly degrades the video quality, since basic RTP-based video streaming *typically* does not involve any means of error recovery. Network jitter and delays result in similar impairments as with voice and include visual artifacts or jerky playback. However, they depend on the concrete error concealment strategy applied by the video decoder.

6.5.1 Approach

We chose three different video clips of various genres as reference. Each video has a length of 16 seconds and has been also used in tests described in Chapter 5. They are chosen to be representative of different kinds of TV content and vary in level of detail and movement complexity. Thus, they result in different frame-level properties and encoding efficiency; *A*) an interview scene, *B*) a soccer match, and *C*) a movie. Each video is encoded using H.264 in Standard Definition (4 Mbps) as well as High Definition (8 Mbps) resolution. Each frame is encoded using 32 slices to keep errors localized. This choice of our encoding settings is motivated by our experiences with an operational IPTV network.

We use VLC to stream each clip with UDP/RTP and MPEG-2 Transport Streams. Without any adjustment, VLC tries to transmit all packets belonging a frame immediately. This leads to traffic spikes exceeding the access network capacity. In effect VLC and other streaming software propagate the information bursts directly to the network layer. As our network capacity, in particular for the access, is limited we configured VLC to smooth the transmission rate over a larger time window as is typical for commercial IPTV vendors. More specifically, we decided to use a smoothing interval (1 second) that ensures that the available capacity is not exceeded in the absence of background traffic. The importance of smoothing the sending rate is often ignored in available video assessment tools such as EvalVid, making them inapplicable for this study. The sequence of frames received at the multimedia client corresponds to the perturbed signal.

We note that Set-top-Boxes in IPTV networks often use proprietary retransmission schemes that request lost packets once (see Chapter 7). Due to the unavailability of exact implementation details, we do not account for such recovery. Our results thus present a baseline in the expected quality; however, systems deploying active (retransmission) or passive (FEC) error recovery can achieve higher QoE as we show in Chapter 7.

We use two different full-reference metrics, PSNR and SSIM, for our quality estimation to compute quality scores from the original and perturbed video streams (see Section 3.3.2). We again remark that PSNR (Peak Signal Noise Ratio) enables the ranking of the same video content subject to different impairments [244, 156]. However, it does not necessarily correlate well with human-perception in general settings. SSIM (Structural SIMilarity) [256] has been shown to correlate better with human perception [258]. We map PSNR and SSIM scores to quality MOS scores according to Table 3.2.

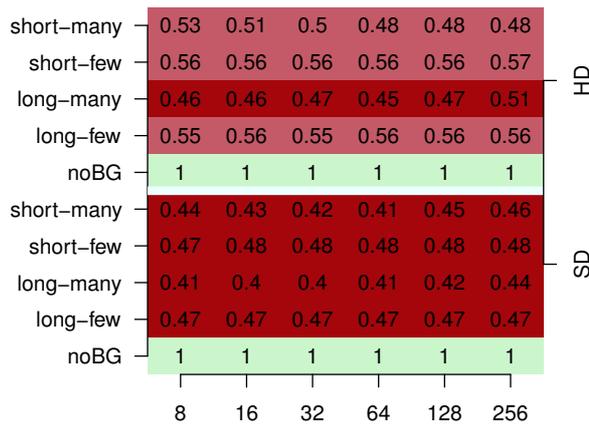


Figure 6.9: Median MOS (color) and SSIM (text) for HD and SD RTP video streams in the access network

6.5.2 Access Networks

We show our results as heatmap in Figure 6.9. The heatmap shows the QoE score for video C sent 50 times per buffer size (x-axis) and workload (y-axis) combination. Each cell shows the median SSIM score and is colored according to the corresponding estimated MOS score (see Figure 6.6(b)); a SSIM score of 1 expresses excellent video quality, whereas 0 expresses bad quality. The upper and the bottom parts of the heatmap correspond to the results of HD and SD video streams, respectively. We omit quality scores obtained for the PSNR metric as they yield predicted scores similar to those obtained by SSIM. Also, as we focus on IPTV where the user consumes TV streams, no video traffic is present in the upstream. For this reason, we only show results for workloads congesting the downlink.

Intuitively, the perceived quality is related to jitter and packet losses, causing artifacts in the video. To show the achievable quality for all buffer size configurations in the absence of background traffic, we show baseline results in rows labeled **noBG**. In these cases, the video quality is not degraded due to the absence of congestion in the bottleneck link.

In the presence of congestion, however, the SD video quality is severely degraded, expressed by a “bad” MOS score (1.5). This holds regardless of the workloads and the buffer configuration; the link utilization by all the workloads cause video degradation due to packet loss in the video stream. We observe that even a low packet loss rate can yield low MOS estimations. Moreover, much higher loss rates (one order of magnitude bigger) can yield the same ratings. For instance, although both scenarios, **long-few** and **long-many**, have a similar SSIM and MOS score for buffers sized to 256 and 8 packets respectively, they show different packet loss rates of 0.5% and 12.5%.

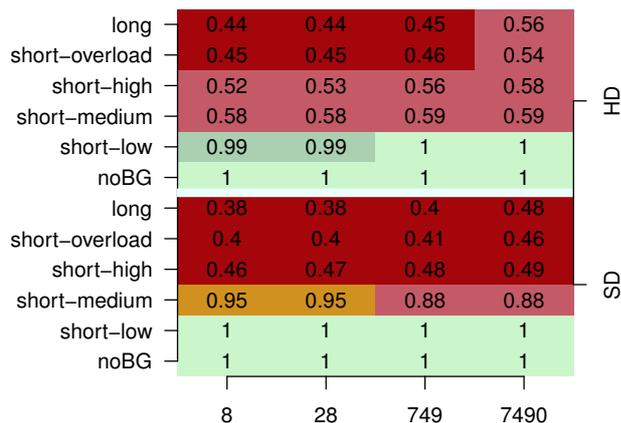


Figure 6.10: Median MOS (color) and SSIM (text) for HD and SD RTP transmission in the backbone

In comparison to the SD video, degradations in HD videos are less pronounced although, in some cases, the packet loss rate is higher. For instance, the packet loss rate for HD and SD video streaming is, with the long-few workload and buffers sized to 256 packets, 2.6% and 1.3% respectively. However, the HD video stream obtains a better MOS score. This interesting phenomena can be explained by the higher resolution and bit-rate of HD video streams, which reduce the visual impact of artifacts resulting from packet losses during video streams.

In the context of the bufferbloat discussion, our results exclude large buffers from being the “criminal mastermind” [3] causing quality degradation, at least for IPTV services. In the case of UDP video streaming in access networks, what matters is the available bandwidth. Moreover, even though buffers regulate the trade-off between packet losses and delay, they have limited influence on the quality from the perspective of an IPTV viewer.

6.5.3 Backbone Networks

Similar to the previous access network scenario, we show the video quality scores obtained for the same video C as a heatmap in Figure 6.10, both for SD and HD resolution. Each cell of the heatmap shows the median SSIM score and is colored according to the corresponding perceptive MOS score (see Figure 6.6(b)). As in the previous scenario, the video was sent 50 times per buffer size (x-axis) and workload (y-axis) configuration. We omit PSNR quality scores as they are similar to the SSIM quality scores.

As in the access network scenario, the bottom row labeled `noBG` shows the baseline results for an idle backbone without background traffic. Similarly, workloads that

do not fully utilize the bottleneck link, i.e., **short-low**, lead to optimal video quality, as expressed by an SSIM score of 1. The reason is that the available capacity in the bottleneck link allows streaming the video without suffering from packet losses.

The first quality degradations are observable in the **short-medium** scenario, where the quality decreases with increasing link utilization. In this scenario, workloads achieve full link utilization for 749 and 7490 packets long buffers more often than for the 8 and 28 buffer configurations. This results in higher loss rates for the video flows, lowering the QoE. This effect is more pronounced for the HD videos which have a higher bandwidth requirement.

Workloads that sustainably utilize the bottleneck link, i.e., **short-high**, **short-overload**, and **long**, yield bad quality scores due to high loss rates. These scenarios provide insufficient available bandwidth to stream the video without losses. Increasing the buffer size helps to decrease the loss rate, leading to slight improvements in the SSIM score.

Comparing the obtained quality scores among the three different videos clips leads to minor differences in quality scores. These differences result from different encoding efficiencies that lead to different levels of burstiness in the streamed video. However, the quality scores of all video clips lead to the same primary observation: quality mainly depends on the workload configuration and decreases with link utilization. Increasing the buffer size generally helps to lower the loss rate and therefore to marginally improve the video quality.

6.5.4 Key Findings for RTP Video QoE

Our results indicate a roughly binary behavior of quality: *i)* when the bottleneck link has sufficient available capacity to stream the video, the video quality is good, and *ii)* otherwise the quality is bad. In between, if the background traffic utilizes the link only temporarily, the video quality is sometimes degraded and sometimes ok. This results in an overall degradation that increases with link utilization. Using HD videos yields marginally better QoE scores even though they use higher bandwidth. This is a result of the smaller visual extent of packet losses. We find that the influence of the buffer size is marginal as delay does not play a major role for IPTV. We did not include QoE metrics relevant for interactive TV or video-calls. Thus, what mainly matters for RTP video streaming is the available bandwidth.

6.6 Web Browsing

We next move to web browsing, our last application under study. To summarize Section 3.3.4, the web browsing experience (WebQoE) can be quantified by two main indicators [71]. One is the *page loading time* (PLT), which is defined as the

difference between a Web page request time and the completion time of rendering the Web page in a browser. Another is the time for the first visual sign of progress. In this study we consider PLT, for which there exists an ITU quality model (i.e., G.1030 [11]) to map page loading times to user scores.

We note that WebQoE does not directly depend on packet loss artifacts, but rather on the completion time of underlying TCP flows. Consequently, factoring in various workloads and buffer sizing configurations—which influence the TCP performance—is particularly relevant for understanding web browsing QoE.

6.6.1 Approach

To evaluate the WebQoE, we map the PLT to a user score z by using ITU recommendation G.1030 [11]. We consider the one-page version of the ITU model, which logarithmically maps *single* PLT's to scores in the range $z \in [1, 5]$ (i.e., 5:excellent, 4:good, 3:fair, 2:poor, 1:bad, as shown in Figure 6.6(b)). This mapping uses six seconds as the maximum PLT, i.e., mapping to a “bad” QoE score. The minimum PLT—mapping to “excellent”—is set to 0.56 (0.85) seconds for access (backbone) scenario, due to different RTTs.

To measure the PLT's, we consider a single static web page, located in one of the testbed servers, and consisting of: one html file, one CSS file, and two medium JPEG images (sized to 15, 5.8, 30, and 30 KB, respectively). The web page is loaded within 14 RTTs, including the TCP connection setup and teardown. Choosing a relatively small web page size was inspired by the frequently accessed Google front page designed to quickly load. To retrieve this web page we use the *wget* tool which measures the transfer time. *wget* is configured to sequentially fetch the web page and all of its objects in a single persistent HTTP/1.0 TCP connection without pipelining. We point out that, as static web pages have constant rendering times, it suffices to rely on *wget* rather than on a specific web browser.

To further analyze the page retrieval performance, we rely on full packet traces capturing the HTTP transactions. We analyze the loss process of the captured TCP flows using the *tcpcsm* tool estimating retransmission events. Further, we measure the RTT during each experiment. PLTs as denoted as RTT dominated if a significant portion of the PLT consists of the RTT component expressed by $14 * RTT$. Similarly, we denote PLTs as loss dominated if the increase in PLT can be mainly attributed to TCP retransmissions.

6.6.2 Access Networks

Figures 6.11(a) and 6.11(b) show heatmaps of the median web browsing quality (MOS) for the access network. Each cell in the heatmap shows the median PLT

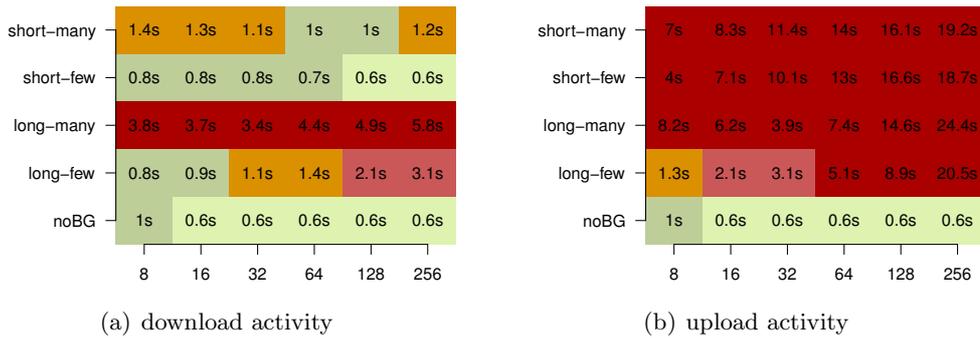


Figure 6.11: WebQoE: Median MOS (color) and page loading times (text)

of 300 web page retrievals per buffer size (x-axis) and workload scenario (y-axis) combination. The heatmap is colored according to Figure 6.6(b).

The baseline results, namely the ones without background traffic, are shown in the bottom row of each heatmap part, labeled `noBG`. The fastest PLT that can be achieved in this testbed is ≈ 0.56 s. As all the cells are green (light gray), we can conclude that in principle each scenario almost supports excellent browsing quality and that any impairment is due to congestion. In this respect, it turns out that, even without background traffic, the WebQoE can be degraded by (too) small buffers, e.g., 8 packets. Due to packet losses causing retransmissions, the PLT is increased to 1 second thereby changing the user perceived quality.

Download activity. Figure 6.11(a) focuses on the scenarios when there is congestion on the downlink. For the `short-few` scenario the downlink is not fully utilized, thus most scores do not deviate much from the baseline results. With this type of moderate workload browsing can benefit from the capacity of large buffers to absorb transient bursts and reduce packet losses. For instance, configuring the buffers size to 256 packets reduces the PLTs to the baseline results (as opposed to PLTs of 0.8s for the smallest buffer configuration). Likewise, for the `short-many` scenario, which involves more competing flows and imposes a higher link utilization, big buffers generally reduce PLTs. As the queuing delays for these scenarios are not excessive, i.e., they are bounded by 192 ms, see Table 6.2, large buffers do in fact improve the end-users perceived quality by limiting the loss rate.

Bufferbloat is visible for the `long-few` scenario, where the median PLT increases with the buffer size, as the PLT is dominated by RTTs caused by large queuing delays. As for the previous scenario, the effects of various buffer sizes are clearly perceived by the end-user (yet in a different manner).

In contrast, the buffer size does not change the WebQoE in the `long-many` scenario. The larger number of competing flows reduces the per-flow capacity and thus the PLT increases beyond the users' threshold of acceptance. As a consequence,

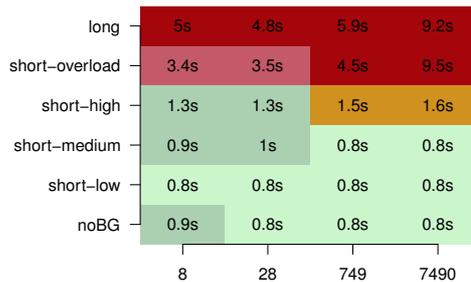


Figure 6.12: Backbone WebQoE

the estimated quality / QoE, in contrast to the previous configuration, can not be improved by adjusting the buffer size. Nevertheless, from a QoS perspective, configuring an appropriate buffer size can allow web pages to load 2 seconds faster. This is not as straightforward since it involves considering the tradeoff between small buffers (packet losses) and large buffers (combined effect of packet losses and large RTTs).

Upload activity. Figure 6.11(b) focuses on the scenarios when there is congestion on the uplink. As expected, congesting the uplink seriously degrades the link overall performance and thereby also the WebQoE. The perceived quality is degraded to the minimum for every buffer size configuration of the scenarios *short-many*, *short-few*, and the *long-many*. The only scenario where the browsing experience will likely be more acceptable is the *long-few* scenario if buffers are small. Such configuration reduces the median PLT from 20 to 1.3 seconds, which maps to a *fair* quality rating.

From a QoS perspective, the figure shows that the PLT and the buffer size are strongly correlated. A wise decision on the dimensioning of the buffers can reduce the PLT from 24.4 to 3.8 seconds (*long-many*). However, and in line with the previous observations, such reductions do not generally suffice to change the user perceived (*bad*) quality.

Combined upload and download activity. In the case of workloads in both, the uplink and downlink direction (not shown), the QoE is dominated by the upload activity. However, due to lower *overall* link utilization and shorter queueing delays (see Section 6.3), the median PLT are less than for the scenarios involving only uploads. The resulting scores generally map to *bad* quality scores; only the *long-few* workload shows better QoE estimates for buffers ≤ 128 packets.

6.6.3 Backbone Networks

The median PLT and the corresponding QoE scores in the backbone setup are shown as a heatmap in Figure 6.12. As in the access scenario, the heatmap shows buffer

sizes on the x-axis and the workload configuration on the y-axis. Each cell is colored according to the MOS scale from Figure 6.6(b) and displays the median PLT of 500 web page retrievals.

The baseline results (**noBG**) show median page loading times of ≈ 0.8 seconds. These loading times are mainly modulated by $14 \times \text{RTT}$ ($\text{RTT} = 60$ ms (see Section 6.2.1)) needed to fully load the page (RTT component), making them higher than in the access network scenarios that has lower RTTs. In this scenario, the distribution of page loading times generally yields a slightly better performance for buffer sizes greater than or equal to the BDP; for these buffer configurations web pages load up to 200 ms faster (80th percentile not shown in the figure). The **short-low** scenario yields similar results despite the existence of background traffic.

We observe the first PLT degradations in the **short-medium** scenario for the 8 and 28 packets buffer configurations. In these cases, PLTs are affected by packet losses causing TCP retransmissions, while the 749 (BDP) and 7490 packet buffers absorb bursts and prevent retransmissions. As in the previous case, web pages load up to 200 ms faster (80th percentile not shown in the figure). The degradations in PLT are, however, small and only marginally affect the quality score.

Degradations in the **short-high** scenario are twofold; while packet losses mainly affect the quality for the 8 and 28 packets buffers, queuing delays degrade the quality for the larger buffers. This effect is more pronounced in the **short-overload** and **long** scenarios that impose a higher link load. In these scenarios, the degradations for the 8 and 28 buffers are mainly caused by packet losses. The 749 and especially the large 7490 buffer affected the flow by introducing significant queueing delays; while the RTT doubles for the 749 buffer configuration, it increases by a factor of 10 for the 7490 buffer. Comparing **short-overload** to **long** for the 8, 28 and 749 buffer size yields a higher number of retransmissions in the **long** scenario, degrading the PLT. With respect to the PLT, short buffers of 8 and 28 packets show faster PLT for the **short-high**, **short-overload**, and **long** scenarios. However, improvements in the PLT do not help to generally improve the quality as the PLTs are already high and lead to bad quality scores.

Our findings highlight the trade-off between packet loss and queueing delays. While larger buffers prevent packet losses and therefore improve the PLT in cases of less utilized queues/links, the introduced queueing delays degrade the performance in scenarios of high buffer/link utilization. In the latter, shorter buffers improve the PLT by avoiding large queueing delays, despite the introduced packet losses. The “right” choice in buffer size therefore depends on the utilization of the link and the buffer.

6.6.4 Key Findings for WebQoE

Our observations fall into two categories: *i*) When the link is low to moderately loaded, larger buffers (e.g., BDP or higher) help to minimize the number of retransmissions that prolong the page transfer time and thus degrade WebQoE. *ii*) When the link utilization is high, however, this increases RTT and thus the page transfers become RTT dominated. Moreover, loss recovery times increase. Therefore, smaller buffers yield better WebQoE despite a larger number of losses.

However, the impact of the buffer size on the QoE metric “page loading time” is ultimately marginal, although the QoS metric “page loading time” sees significant improvements. While this may seem surprisingly counterintuitive at first, let us consider a twofold improvement of the page loading time from 9 seconds to 5 seconds. This improvement is large for the QoS metric, but it is insignificant for the QoE metric, as both 9 and 5 seconds map to “bad” performance.

6.7 Excursion: YouTube / HTTP Video Streaming

In this excursion, we present a first step towards the study of HTTP Video Streaming in a real-life context. In contrast to IPTV and other UDP/ RTP based streaming services, HTTP based streaming uses TCP for reliable data delivery. As a consequence, neither video nor audio suffer from impairments caused by lost data. Impairments are solely introduced by the application level buffer running empty in situations when the incoming bitrate sustains to be lower than the playback bitrate. This leads to rebuffering events, which alter the temporal structure of the video and result in stalling events—noticeable by jerky playback or waiting—that degrade the QoE. In this excursion, we focus on YouTube as major HTTP video streaming platform. In the evaluation, we focus on backbone networks and leave the extended evaluation including access networks for future work.

6.7.1 Approach

Our evaluation relies on using the actual YouTube CDN infrastructure for video delivery. A peculiarity of YouTube is that a video is delivered within a single and bursty TCP connection: the first 30 to 40 seconds are sent as bulk transfer to quickly fill up the application layer buffer, whereas the remaining transfer is throttled and sent in smaller bursts [99].

We conduct the YouTube QoE evaluations as follows. One of our multimedia client nodes (see Figure 6.3) runs an instance of the Opera browser and fetches YouTube content via a proxy hosted at the opposite end of the topology. The proxy ensures that all traffic directed to YouTube servers traverses the bottleneck link. The client-side video playback is performed by the Flash version of the original YouTube

| Video clip | Length [sec] |
|-------------------------|--------------|
| Avatar movie trailer | 211 |
| The Dark Knight trailer | 150 |
| User supplied content | 71 |

Table 6.3: Selected YouTube video clips

player running in an instance of the Opera browser. Its graphical output is rendered to a virtual X11 frame buffer. A fresh copy of the browser is used for every video playback, enforcing empty caches and the same initial browser state for every video.

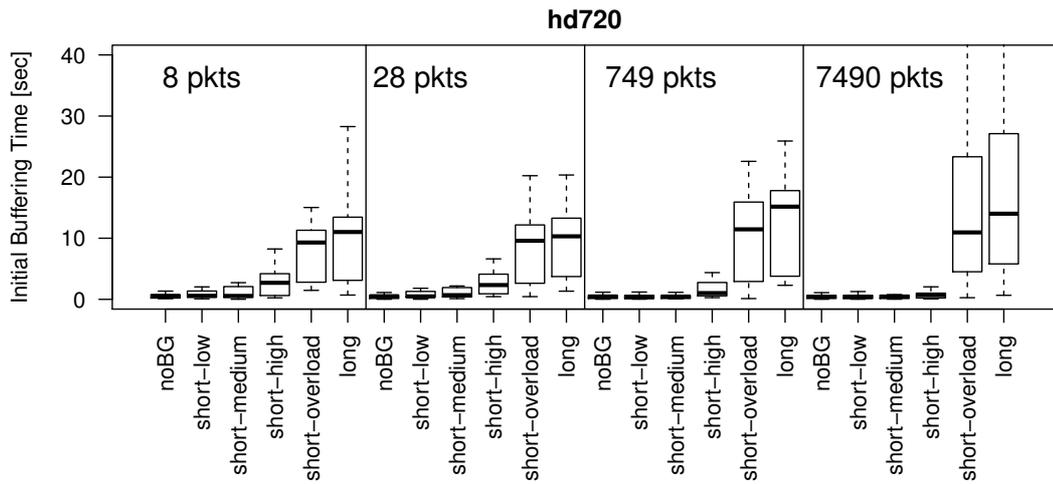
For both the backbone and access network setups we consider three video clips, as shown in Table 6.3. In the backbone setup each movie is played back at all five available quality settings: small (240p), medium (360p), large (480p), hd720 (720p), and hd1080 (1080p).

6.7.2 Backbone Networks

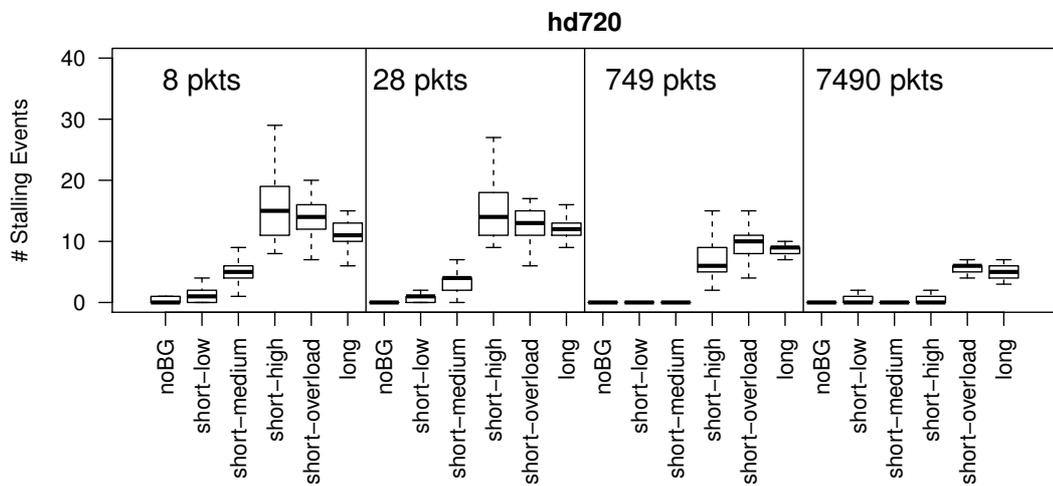
We evaluate the QoE of HTTP video streaming subject to temporal stimuli caused by network layer impairments. As no complete QoE model for the assessment of HTTP video exists, we base our evaluation on quality indicators reported in the literature. These indicators provide insights in factors impacting QoE rather than allowing for a mapping to QoE scores. We assess the impact of buffer sizing on the number of stalling events and their respective lengths, which are the two main quality indicators for HTTP video streaming [124, 179]. Another QoE indicator, is the initial buffering time denoting the time to fill up the application buffer and to start the playback. This is however less relevant, as users prefer waiting longer for the video to start rather than watching a jerky playback [125].

We show the results for the three quality indicators (initial buffering time, number of stalling events, and stalling event duration) in Figure 6.13 for HD720p resolution. We omit other resolutions as they show similar behaviour. Each figure shows four cells separated by vertical bars, each corresponding to the buffer size configurations of 8, 28, 749, and 7490 packets. Each cell shows results for the six workload configurations (**noBG** to **long**) ordered by increasing link utilization. The baseline results, i.e., the ones without background traffic, are labelled **noBG**. Recall that we access external servers and stream videos from the YouTube CDN. Therefore, the baseline results already contain a slight degree of variability that is caused by external networks beyond our control. We additionally checked that no advertisement were played back, as they would bias our measurements.

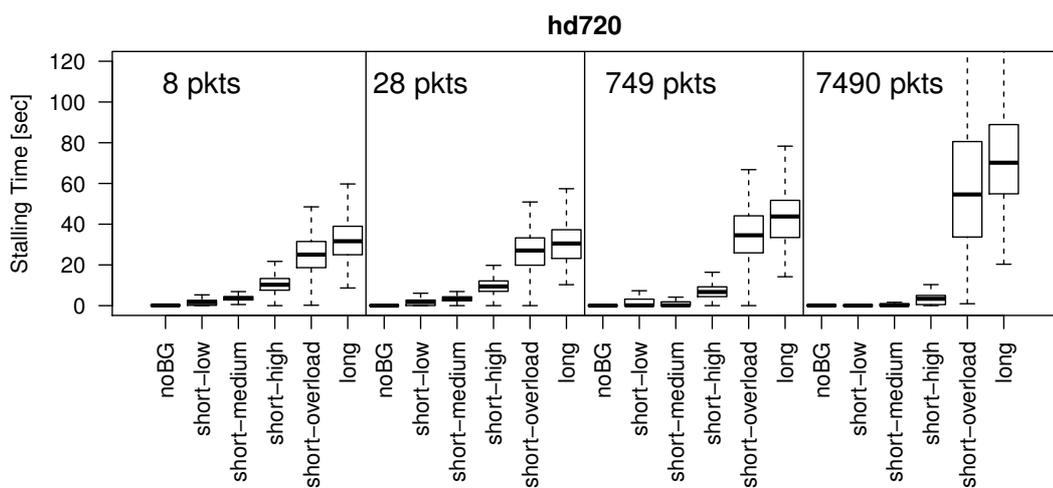
The YouTube results shown in Figure 6.13 are in line with our previous results, i.e., workload rather than buffer size is the primary QoE determinant. In general,



(a) Initial stalling time



(b) Number of stalling events



(c) Stallingtimes

Figure 6.13: YouTube quality indicators for HD720 resolution

low values denote higher QoE. In particular, Hossfeld [124] showed that 1 to 3 seconds stalling length and 1 to 3 stalling events can already yield low MOS values < 3 . These values are almost always exceeded in high load regimes (e.g., short-high, short-overload, and long).

With increasing link utilization, the YouTube application buffer takes longer to fill and the initial buffering time increases. As a result, the user has to wait longer before the video starts to play. Due to decreasing goodput, the number of stalling events and their stalling times also increase with increasing link load. While increasing the buffer size reduces the number of stalling events, their respective lengths increase.

The best QoE was observed for low to medium loaded links (noBG, short-low, and short-medium) for buffers configured to the BDP or larger. This finding is in line with our earlier findings for web QoE. Recall that when the link is low to moderately loaded, larger buffers (e.g., BDP or higher) help to minimize the number of retransmissions and thus increase the goodput that is required to play the video without stallings. When the link is utilized, however, the stream suffers from queueing delays induced by large buffers and the goodput decreases. While this leads to less stalling events, stalling times increase.

In summary, a certain amount of buffering is required to reduce the amount of packet losses. High link loads primarily cause stalling events that will detrimentally effect QoE.

Our results can be generalized to a wide range of similar streaming services such as video on demand services offered by TV stations. We note that our results arguably also cover the baseline performance of Netflix, a major subscription-based VoD service in the US and Canada. Unlike Youtube, Netflix uses a seamless rate adaption mechanism to switch between quality profiles during streaming [28], while our study instead focuses on a fixed quality profile in terms of the underlying progressive download technology used by YouTube.

6.8 Related Work

The rule-of-thumb [136, 250] for dimensioning network buffers relies on the bandwidth-delay-product (BDP) $RTT * C$ formula, where RTT is the round-trip-time and C is the (bottleneck) link capacity. The reasoning is that, in the presence of *few* TCP flows, this ensures that the bottleneck link remains saturated even under packet loss. This is not necessary for links with many concurrent TCP flows (e.g., backbone links). It was suggested in [250] and convincingly shown in [35, 45] that much smaller buffers suffice to achieve high link utilizations. The proposal is to reduce buffer sizes by a factor of \sqrt{n} as compared to the BDP, where n is the number of concurrent TCP flows [35]. Much smaller buffer sizes have been proposed, e.g., drop-tail buffers with $\approx 20 - 50$ packets for core routers [76]. However, these come at

the expense of reduced link utilization [45]. This problem has been addressed by a modified TCP congestion control scheme that aims to maintain high link utilizations in small buffer regimes [109]. For an overview of existing buffer sizing schemes we refer the reader to [251].

While the above discussion focuses on backbone networks, more recent studies focus on access networks, e.g., [70, 158, 174, 240], end-hosts [3], and 3G networks [112]. These studies find that excessive buffering in the access network exists and can cause excessive delays (e.g., on the order of seconds). This has fueled the recent bufferbloat debate [16, 98] regarding a potential degradation in Quality of Service (QoS).

Also, buffer sizing has been considered in the context of green networking. In this context, [252] proposes to dynamically disable unused buffer space to reduce the energy consumption of routers. This is an optimistic proposal, as the overall memory installed on a linecard, including the main memory needed to run the OS, only accounts for 13% of the line card's power consumption [263].

Indeed, prior work has shown that buffer sizing impacts QoS metrics. Examples include *network-centric* aspects such as per-flow throughput [208], flow-completion times [164], link utilizations [45], packet loss rates [45], and fairness [254]. Sommers et al. studied buffer sizing from an operational perspective by addressing their impact on service level agreements [233]. However, QoS metrics and even SLAs do not necessarily reflect the actual implications for the end-user. Thus, in this chapter, we present the first *user-centric* study of the impact of bufferbloat and background traffic by focusing on user perception as captured by QoE models and quality metrics.

6.9 Discussion

The goal of this chapter is to elucidate the open problem of proper buffer sizing and to pave the way for more informed sizing decisions. In this respect, this chapter presents the first comprehensive study of the impact of buffer sizes on *end-user experience*. This is a highly relevant perspective since it has implications for network operators and service providers, and by extension, device manufacturers.

To tackle this problem, we first evaluate the impact of buffering in the wild using a large data set from a major CDN that serves for a large number of Internet users (80M IPs from 235 countries and location identifiers according to the Maxmind geolocation database). Our analysis shows that buffering is likely to be prevalent on a large scale. This motivates our further evaluation of buffer sizing.

The main contribution of our work is an extensive investigation of the impact of buffer sizing on user Quality of Experience predictions. Specifically, we use a testbed-driven approach to study three standard application classes (voice, video, and web)

in two realistic configurations emulating access and backbone networks. Our evaluation considers a wide range of traffic scenarios and buffer size configuration, including very large buffers.

Our main finding is that the *level of competing network workload* is the primary determinant of user QoE. While buffering has a significant impact on QoS metrics, it only marginally impacts QoE / quality metrics. This leads us to conclude that limiting congestion, e.g., via QoS mechanisms or over-provisioning, may actually yield more immediate improvements in QoE than efforts to reduce buffering. There are, however, several subtle issues that complicate buffer sizing.

The application and the level of congestion determine the potential impact of buffer size choices. For instance, in the case of Web browsing, large buffers yield better QoE for moderate network loads, while smaller buffers improve QoE for high network loads. Despite the potential for optimization, the impact of reasonable buffer sizes on QoE metrics is marginal. While such findings may be regarded by some as ‘unsurprising’, it is important to recognize that we have presented the first study to provides a quantitative assessment of buffer sizing and user QoE / quality. This is of particular importance for network operators, as it indicates that as long as buffers are kept to a reasonable size their impact is of marginal relevance.

With respect to the ongoing bufferbloat debate, our main claim is that only relatively narrow conditions seriously degrade QoE, i.e., when buffers are oversized and sustainably filled. Such conditions indeed occur in practice, as our empirical evaluation and other recent studies confirm, but their occurrence is relatively rare. Therefore, the ongoing efforts of the bufferbloat community to drive engineering changes and to advocate new AQM mechanisms appear to be rather precipitated than based on solid evidence.

6.10 Future Work

Our study leads to a basic understanding of the impact of buffering on QoE / quality. Future work should extend this understanding in the following directions. The first direction consist in getting an understanding of how buffering in the access changes the traffic pattern (e.g., traffic burstiness) in the core, in particular in the case of excessive buffering (bufferbloat) at the edge. This burstiness can have implications on the buffering that is required in the core, in particular as all-optical networks equipped with small buffers could become reality in 10+ years from now.

The second direction should extend the presented analysis by investigating online games as delay sensitive application. This can be evaluated automatically in an objective study by letting bots play the game and by measuring their performance subject to different buffer size and traffic configurations. Thus, the evaluation setup should consist in operating a game server and a game client in the testbed. The

game client should be able to perform automated tasks, e.g., to measure the task completion time. Preliminary work shows the feasibility of this study by using the game Minecraft (see the master's thesis by Hannes Fiedler [89]).

The third direction should extend the presented video streaming analysis by the application of more advanced QoE metrics for RTP and HTTP video streaming (when available). One particularly interesting direction is the study of adaptive HTTP video streaming (e.g., DASH).

The fourth direction consists in understanding the impact of active queuing mechanisms on QoE. Such mechanisms aim at regulating (TCP) traffic by dropping packets before the buffer capacity is exceeded. This is particularly relevant as CoDeL [188] is currently advocated as new active queue management algorithm to fix bufferbloat. The consequences of devices deploying CoDeL and its potential to fix bufferbloat are not yet well understood, in particular not from a QoE perspective.

Lastly, as our findings highlight utilization and traffic dependent quality impacts, our results should pave the way for a traffic-dependent dynamic buffer sizing scheme.

Part II

Impact of Application Buffer

7

QoE Impact of Retransmission Buffers in IPTV Set-Top-Boxes

The provisioning of broadband access has enabled the deployment of new services such as the distribution of TV content over IP networks (IPTV) or Video on Demand platforms. Recall that for the successful deployment of such services, it has become increasingly important for service providers/operators to understand and control Quality of Experience (QoE) aspects. The common solution adopted by service providers consists in the real-time monitoring of QoE. Observations of drops in the QoE levels demands for network control to improve the service delivery and ultimately the customers' satisfaction.

For running in real-time, monitoring solutions face the critical requirement of relying on QoE models of low to moderate computational complexity. This requirement is met by parametric QoE models (see Section 3.3) which rely on easily measurable QoS and content parameters (e.g., packet loss, jitter, bitrate) and simple heuristics for QoE computation. However, QoE parametric models are mostly limited to measures captured in the core or access network and may thus neglect the QoE impact of recovery mechanisms deployed at client-side. This includes recovery mechanisms that are usually deployed on various layers in the protocol stack. For instance, on the application layer, visual concealment techniques are commonly used for video loss recovery. Without considering such recovery mechanisms, QoE models are prone to mispredict QoE and consequently may lead to suboptimal network control by the

⁰The content of this chapter is joint work with Balamuhunthan Balarajah, Sebastian Benner, Alexander Raake, and Florin Ciucu. It is published at the IEEE International Conference on Communications [118] in 2011.

operators. In order to prevent such misguided reactions, it becomes critical for the operators to understand the influence of recovery schemes on QoE models.

To better explain the scope of this work, we remark that recovery mechanisms are often categorized into *active* and *passive*. Passive recovery techniques can be implemented at either the transport or application layer by using FEC algorithms to embed redundant information in the bitstream and therefore allow for bit-exact error recovery without any interaction between sender and receiver [221, 137]. Such characteristics make passive recovery appropriate for delay critical interactive real-time applications, e.g., VoIP or video conferencing. In turn, active recovery techniques—such as Automatic Repeat Requests (ARQ)—are characterized by the interaction between the sender and receiver in the form of retransmission requests of lost information. Moreover, they induce much lower network load and processing overhead at low error rate and are thus preferred for error control design.

In order to address the problem of underestimating QoE due to neglecting recovery mechanisms, we study in particular the active recovery mechanism that is implemented in the Microsoft TV (MSTV) solution used by several big ISPs (e.g., Deutsche Telekom, AT&T) for their IPTV systems. MSTV implements active recovery at client-side in the Set-Top Box connected to the TV set. By identifying aspects of the buffering behavior and the loss recovery mechanism implemented in the STB, we want to pave the way for more informed QoE models that account for error recovery at client side.

Our study is challenged by the unavailability of ARQ implementation details of the STB, which are proprietary. In order to circumvent this problem, we follow a two step approach. In the first step, we measure the behavior of the MSTV STB when exposed to a range of different network conditions (packet loss and jitter). We then compare the empirically observed behavior to simulations of three ARQ algorithms of different complexities and evaluate their efficiency in recovering errors and the amount of generated overhead. Such factors are particularly important for the ISP as they contribute to the achievable QoE. Based on empirical STB observations and further comparisons with simulation results of our ARQ algorithms under realistic network conditions, we are able to provide insights, into the ARQ scheme implemented in a widely used IPTV system and its impact on QoE. Moreover, the observed results lead us to speculate that MSTV uses simple ARQ schemes which are sufficient to drastically improve the QoE.

7.1 Measurement Setup

In order to study the behavior of the resend (ARQ) scheme implemented in the Microsoft TV (MSTV) Set-Top-Box (STB), we use an edge-based measurement setup as depicted in Figure 7.1. This scenario resembles a home setting with an STB connected to an ADSL line subscribed to the IPTV service. The considered scenario

accounts for congestion in the access network where multicast and ARQ traffic is impaired by the same loss process. However, due to the considered topology, it does not account for different loss processes between ARQ and multicast traffic in backbone networks which may arise, e.g., in the presence of different routes. The study of this setting is interesting as errors are known to mostly occur in access networks which form a bottleneck and rarely in the (mostly) overprovisioned and/or QoS enabled backbone network.

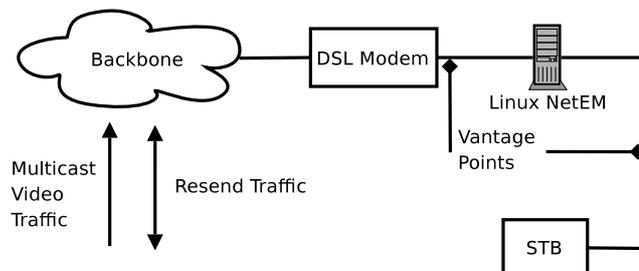


Figure 7.1: Measurement setup

To make our setup and subsequent analysis more realistic, we account for different network conditions by injecting uniform and bursty packet losses with and without jitter. To this end we use a Linux machine running the Network Emulator (NetEm) [116, 140] functionality. NetEm is the standard network emulator in the Linux kernel. It provides methods for introducing delay and jitter, packet loss, packet reordering, and packet duplication. Its functionality is similar to Dumynet [1]—the standard emulator in Free BSD and OS X—and NIST Net [4]—which is no longer maintained and has been largely incorporated into NetEm. Due to limitations of NetEm in supporting the Weibull distribution, the injected jitter is generated instead according to the normal distribution.

When switching to a TV channel, the STB joins a multicast group to receive a TV channel at a constant bitrate of 4 Mbps (8 Mbps) for SD (HD) resolution, respectively. The measurement setup uses a capture process to automatically record a large set of traffic traces with and without impairments¹ over multiple days. This is accomplished by placing the vantage points before and after NetEm. The capturing process works as follows. Once the network impairment is configured in the emulator we wait for 20 seconds to allow the STB to stabilize and then start the capturing process for 130 seconds. Upon the completion of the capture process we reset the loss and jitter settings in the emulator to provide the STB with an additional 20 seconds of unimpaired traffic before capturing the next network impairment configuration. The obtained traffic traces enable the empirical analysis and sets the basis for simulating the STB behavior.

¹We remark that the incoming multicast stream captured before NetEm can already contain packet losses. We exclude such errors in our analysis, as the unimpaired network stream is not known. Instead, we focus on impairments added in a controlled fashion by NetEm.

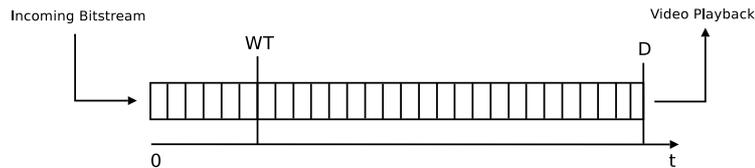


Figure 7.2: De-Jitter buffer model

7.2 Simulation Design and Buffer Models

We remark that implementation details of the STB are proprietary and thus not known. To reveal insights into the functioning of the resend mechanism implemented in the STB, we designed a discrete event simulator to evaluate the behavior of the setup in Figure 7.1. The simulator only uses the multicast traffic contained in the captured traces by isolating the additional ARQ traffic. This isolation replicates the network conditions faced by the STB, including the packet loss and jitter processes. In this way we are able to appropriately speculate about the ARQ design used in the STB.

7.2.1 De-Jitter/Playout Buffer Model

An important component of our simulator is the de-jitter/playout buffer. This component is generally featured by multimedia applications in the Internet where the packet delay is known to fluctuate rapidly (jitter) [46] due to processing and queuing delays. In order to compensate for such varying network delay, multimedia applications generally deploy de-jitter buffers that delay the playout of a packet by buffering at the client-side. In general, such buffers can be implemented in a static or dynamic manner [218, 239], depending on whether the buffer depth is fixed or dynamically adapted during streaming.

De-jitter/playout buffers are especially necessary in the presence of ARQ mechanisms in order to wait for the generated retransmissions to arrive before their scheduled playout time. The time needed for the waiting process depends not only on the conditions of the network (RTT and jitter), but also on the processing delay of the server hosting the retransmission cache and handling the retransmissions. Due to such factors, the dimensioning of buffer is generally a challenging problem.

For the sake of simplicity, we consider the static de-jitter buffer illustrated in Figure 7.2. The buffer holds D ms of incoming traffic and serves the video decoder with a constant rate bitstream of packets that are free of jitter. The playout of each packet is delayed for D ms due to buffering. Packets which never arrive by their scheduled playout time are considered to be lost and discarded. Note that for our

simulation settings we use a value of $D = 1000$ ms for the buffer depth, suggested by observations in the measurement results.

7.2.2 Implemented ARQ Schemes

In addition to the de-jitter buffer, our simulator implements three ARQ schemes of different complexities. The first two mechanisms are motivated by how the Real-Time Control Protocol (RTCP) [13] defines feedback in RTP streaming systems. The third scheme is motivated by the STB behavior observed in measurements.

Each ARQ scheme triggers a packet loss detection after waiting for WT time, in order to allow for out-of-order packets to arrive before the loss detection. Estimates of WT and the mechanism used by MSTV are provided in Section 7.3.4. In future work, we plan to study the optimal range of WT .

We assume that the deployed transport protocol implements sequence numbers such that any missing packets can be easily detected by gaps in the received packet sequence. This is realized by the Real-time Transport Protocol (RTP) that is used for streaming in the studied IPTV system. Packets that are missing at time WT —relative to the beginning of the buffer—are considered to be lost and are subsequently requested by the ARQ protocol. In order to increase fault tolerance, all three mechanisms issue two identical copies of the same request packet spaced randomly 10 to 30 ms from each other, as also observed from the STB measurements (see Section 7.3.1).

In the following we give a more detailed description of the implemented ARQ schemes starting with the simplest one.

Mechanism 1 In this scheme a resend-request is generated for each lost packet. As mentioned earlier, this elementary approach is motivated by the immediate feedback mode discussed in [13], but is however restricted to allowing for a single request per packet only.

Setting $WT = 0$ results in the situation in which the lost packets are immediately requested when the next packet is enqueued in the buffer. An undesirable consequence of this extreme setting is that it triggers an increase of the network load and of the retransmission cache server in case of out-of-order packets. Setting higher waiting times WT , within the buffer depth D , allows out-of-order packets to arrive before requests are issued.

We point out that, in general, $WT < D - \text{RTT} - \sigma_{\text{Jitter}}$ must hold in order to allow retransmissions to arrive before playout (here, RTT denotes the mean delay to the resend-server and σ_{Jitter} denotes the standard-deviation of the delay variability).

Mechanism 2 To avoid the inherent increase in network load characteristic to *Mechanism 1*, we now allow for requests of multiple video packets in a single compound request. This is implemented in the resulting *Mechanism 2* by a timer that is started at the first packet loss and expires after WT time. All packet losses that occur during this time period are requested in a batch. After the resend request is sent, the timer is stopped and is started again upon the next packet loss event.

Mechanism 3 This mechanism allows for the periodic generation of resend requests as observed in the STB measurements (see Section 7.3.1. Concretely, such a request is issued every WT time, iff $n \geq 1$ packet losses occur within WT time. This is implemented by modifying the timer introduced in *Mechanism 2* to start when the first packet enters the buffer and to expire after WT time. The timer restarts irrespectively of whether a request was issued or not.

7.3 Evaluation of Set-Top Box Behavior

In this section we discuss the empirical measurements and simulations of the ARQ schemes. Our main objective is to gain insights into the properties of the ARQ scheme implemented by the MSTV STB. We start by first analyzing the observed STB behavior. Then we proceed to compare the measured STB performance and corresponding simulations with respect to correction efficiency and induced network load. The observed STB correction efficiency is further analyzed in terms of achievable QoE improvements.

Our study focuses on STB service degradations which can provide direct insight into the details of the implemented ARQ scheme. We particularly evaluate the statistical properties of the request and retransmission packets from the capture traces, and which are next discussed.

7.3.1 Request Packets

We start our analysis by investigating properties of retransmission requests issued by the STB. For those requests we observed that the STB sends two identical packets. As pointed out earlier, the approach of requests duplication increases the fault tolerance by increasing the probability of retransmission success.

We also evaluated the delay between two identical copies of the same request and between two non-identical requests. The corresponding distributions are shown in Figure 7.3.(a) and (b). In the case of identical copies, the distribution shows peaks at $\{1, 10, 20, 30, 60, 90\}$ ms, where 30 ms was found to be the dominant peak in all of our measurements (see Figure 7.3.(a)). In turn, in the case of non-identical requests, the delay distribution shows a dominant peak at 120 ms (see Figure 7.3.(b)).

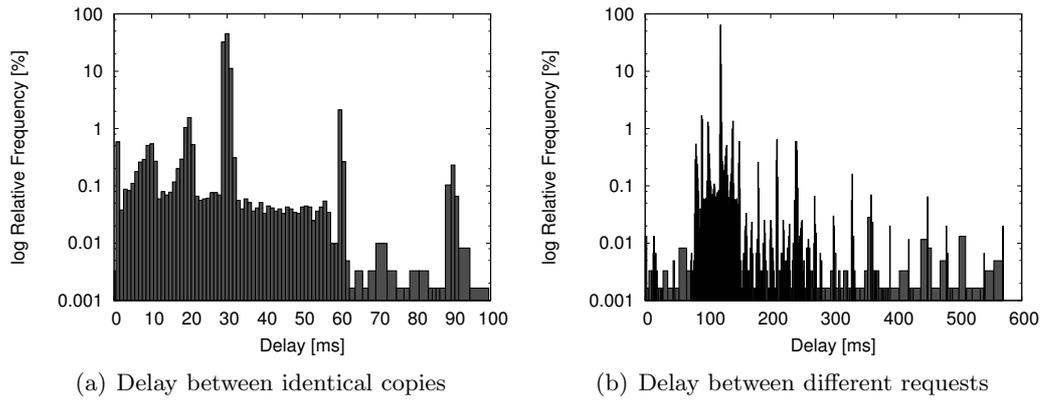


Figure 7.3: Delay between requests for the uniform loss dataset

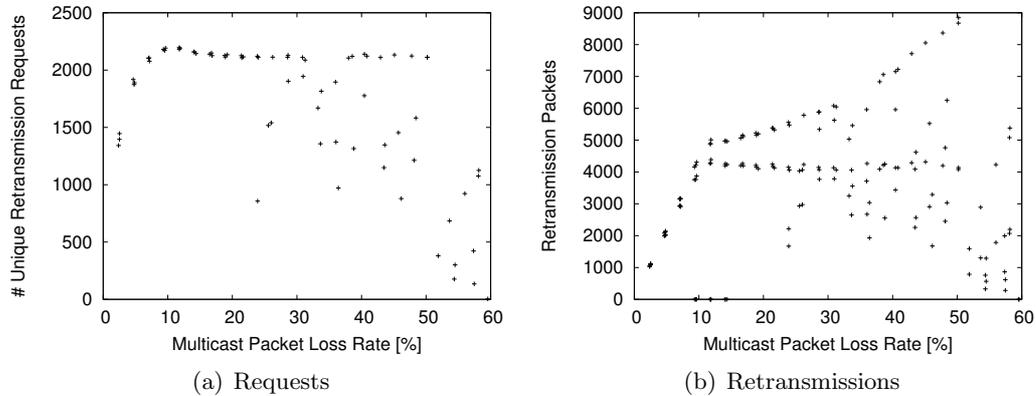


Figure 7.4: Number of request and retransmission packets for the uniform loss dataset with traces of length 130 sec

In addition to the delay distributions, we observed a rate of no more than 10 non-identical requests per second issued by the STB per second, as shown by the upper bounded traffic in Figure 7.4.(a). This indicates the presence of periodic checks in the STB which motivated our Mechanism 3.

Moreover, in terms of packet sizes, the analysis reveals that only 4 distinct sizes are used, i.e., 151, 154, 157, 160 bytes for the UDP datagram length. By increasing the loss rate we observed an interesting tendency for higher packet sizes. The observed packet sizes suggest the existence of multiple slots used for requesting packets; recall that the elementary *Mechanism 1* uses a single slot. To better understand whether slots are used in STB for single or multiple requests we analyzed traces with periods of subsequent losses. As no trend of increasing packet sizes was observed, as opposed to the case of isolated losses, we conclude that a slot is used by STB for requesting loss bursts consisting of subsequent packets.

7.3.2 Retransmission Packets

Unlike the original RTP packets which are sent to the STB by multicast, the corresponding retransmission packets are sent by unicast. This implementation detail permits the identification of the loss packets as the format used for requests is not known.

The total number of retransmissions per packet loss rate from our measurements is shown in Figure 7.4.(b). The figure indicates the presence of two trends. The first trend shows a saturation for isolated losses at $4 \cdot 10^2$ packets per second at loss rates $> 10\%$. In turn, the second trend shows a steady increase in the number of retransmissions for loss bursts. Unlike the case of retransmissions, the amount of request packets is always upper bounded. This observation supports our previous speculation on the usage of slots in request packets by the STB.

7.3.3 Comparing Correction Efficiency

Correction efficiency of the deployed ARQ mechanisms is particularly important for the achievable QoE at the client side: the more losses can be corrected before playout the higher the resulting visual quality. To understand the correction efficiency, in the particular case of the STB, we evaluate it for several broad network conditions with multiple (WT, packet loss) combinations under uniform and bursty packet loss with and without jitter. For every (WT, packet loss) combination we run 1000 simulations and show the average correction efficiency. The results are compared to the measured STB behavior which is constant for every (WT, packet loss) combination.

Figure 7.5 shows the results for uniform packet loss without jitter under unlimited slots for *Mechanisms* 2 and 3. For small values of WT, the buffer provides sufficient buffering to allow all retransmissions to be completed. Thus, both the simulated algorithms and the STB implementation achieve complete loss correction except for lost requests and retransmissions. We also observed (figure not shown here) that by limiting the number of slots to 4, *Mechanisms* 2 and 3 perform slightly weaker than for the default case of unlimited slots.

In contrast to the case of small values of WT, the observed correction efficiency optimal behavior decays at larger values of WT which are closer to the chosen buffer depth of one second. The reason is that retransmissions can arrive after the scheduled playout and are thus discarded.

In the case of bursty packet loss and jitter we observed similar correction efficiency behavior (figures not shown here). As a direct consequence for accounting for jitter, small delay variations (e.g., mean = 100 ms) do not influence the correction efficiency at small values of WT.

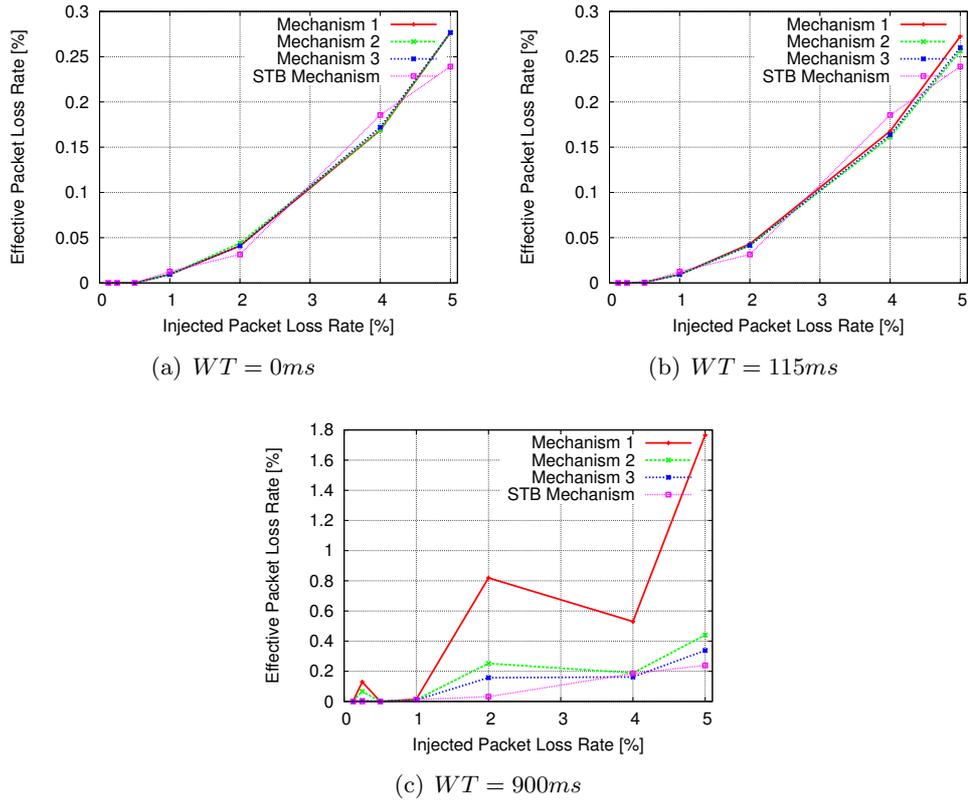


Figure 7.5: Correction efficiency for uniform packet loss

7.3.4 Comparing Induced Network and Server Load

In addition to providing desirable QoE to its customers, service providers are also considered about the influence of the deployed mechanisms on the network and server load. In this section we express and evaluate this load by the amount of request packets generated by the different ARQ mechanisms and the STB, under the presence of both loss and jitter.

Figure 7.6 shows the generated retransmission traffic when the video transmission is impaired by uniform packet loss. In the extreme case when no waiting time is specified ($WT = 0$, see Figure 7.6 (a)), all considered mechanisms generate significantly more load than the STB when the packet loss is greater than .5%. Moreover, *Mechanisms 2* and *3* barely show any improvement in decreasing load over *Mechanism 1*, because of infrequent grouping multiple requests in one packet in the particular case of $WT = 0$. The visible slight improvement can be explained by the existence of loss bursts. In another extreme case of high WT , both *Mechanisms 2* and *3* generate significantly less load than the STB, whereas *Mechanism 1* does not change its behavior from the case of $WT = 0$. In contrast to these differences, we observed however that *Mechanism 3* with an unlimited number of slots performs similar as the STB when WT is in the order of 100 ms (figure not shown here). These observations lead us to speculate that the STB implements an ARQ scheme similar to *Mechanism 3* with $WT \approx 100$ ms.

When jitter is also present, in addition to loss, we further observed that the load increases at small values of WT (see Figure 7.7) for all *Mechanisms* and the STB. The explanation stems from the request of out-of-order packets. However, at higher values of WT , only *Mechanism 1* and STB experience a load increase. This indicates that the STB behavior is sensitive to the presence of jitter than loss.

7.3.5 QoE Impact

To analyze the impact of ARQ on QoE as predicted by parametric QoE models, we used a parametric model specifically tailored to the MSTV system [94]. We applied the model both to the measured loss process before and after ARQ. In this way, we can quantify the QoE improvements achieved by the ARQ scheme implemented by the MSTV STB.

The QoE improvement for uniform packet loss is schematically shown in Figure 7.8 for an 16 Mbps HD video. The obtained results show an expected drastic improvement of the QoE in the presence of ARQ. However, the achievable QoE improvement depends on the loss process obtained after applying ARQ and will thus vary with different ARQ schemes. In practical monitoring, parametric QoE models can be combined with analytical models describing the correction efficiency of the used ARQ scheme or augmented with live measurements of ARQ traffic.

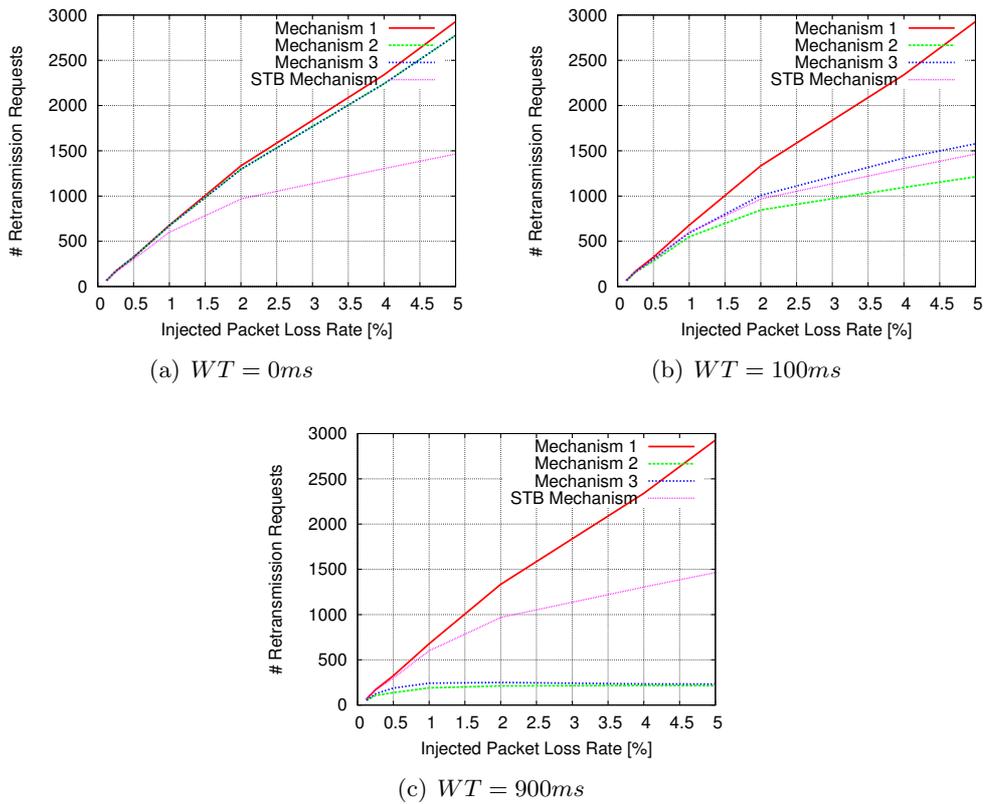


Figure 7.6: Network and server load for uniform packet loss

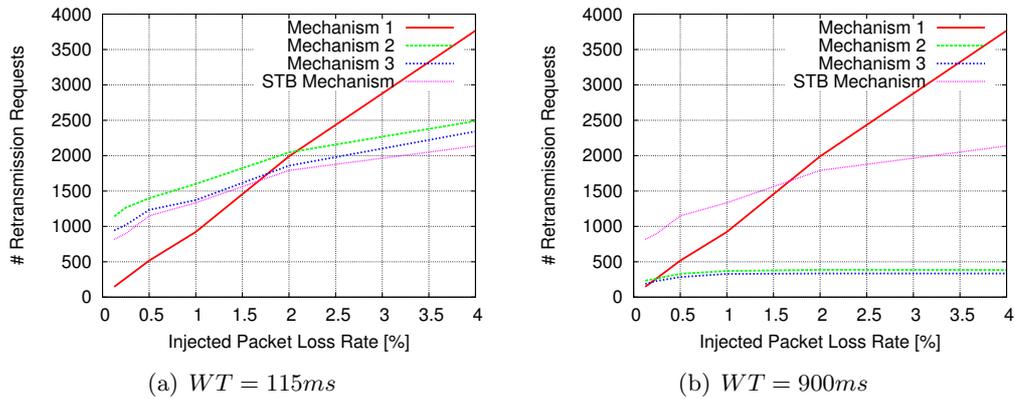


Figure 7.7: Network and server load impact for uniform packet loss and normally distributed jitter with a mean=10ms

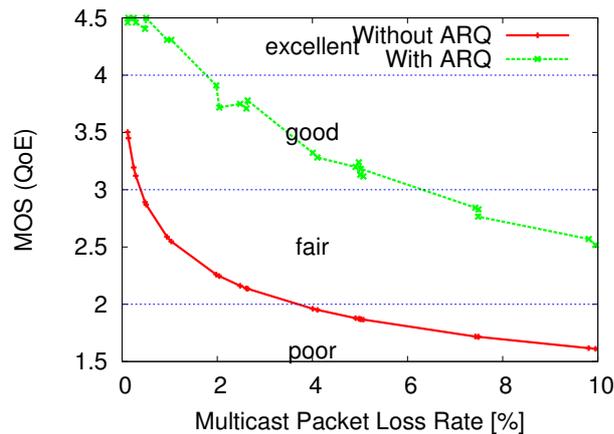


Figure 7.8: QoE with and without MSTV ARQ

7.4 Related Work

The performance of ARQ in the fast RTP retransmission scheme was evaluated in [210] based on simulations. This study discusses the amount of caching required at ARQ servers and the influence of multiple retransmission requests to increase fault tolerance. Our study complements this work by evaluating multiple mechanisms and comparing the obtained results against the behavior of a real STB. To the best of our knowledge, we are the first to analyze the behavior of the MSTV STB. Additional, but less related works, include Jiang and Schulzrinne [138] which studies the efficiency of passive repair techniques subject to bursty packet loss impairing audio transmissions. The performance of FEC was studied analytically for multimedia streaming in more general settings in [93]. The packet loss introduced by a de-jitter buffer due to discarding out-of-order packets was analytically bounded in [62] for static de-jitter buffers. In addition, [184] derived the buffer depth to meet a certain packet loss rate by discarding late arrivals. The amount of buffering necessary for a smooth playback was analytically studied in [195]. Pessimistic performance bounds for FEC and ARQ were derived in [276].

7.5 Discussion

We empirically evaluated the performance of the ARQ mechanism deployed in the MSTV Set-top Box (STB) used by several big ISPs. As concrete implementation details are proprietary, we performed a reverse engineering study. Such a reverse engineering study is challenging as internals of the STB (e.g., byte code on its hardware) were inaccessible to us, turning the STB into a black box that can only be studied through measuring its behavior. Any reasoning on the implemented ARQ

scheme is thus purely based on the observed *encrypted* network traffic generated by the STB subject to network impairments such as loss and jitter. The inability to decrypt the generated traffic renders the study of packet payloads infeasible. Access to the payload would, however, have simplified the reverse engineering effort.

Our reverse engineering study followed a two-step approach. In a first step, we studied empirical measurements of the STB behavior subject to packet loss and jitter. In a second step, we simulated three ARQ schemes of different complexity and compared their performance to the observed STB behavior. The ARQ performance was evaluated under different network conditions: bursty and uniform packet loss with and without jitter. This evaluation led us to speculate on the implemented ARQ mechanism.

We concluded that the STB implements a rather simple ARQ algorithm which is sufficient for maintaining desirable QoE levels. The application of a parametric QoE model showed a drastic improvement of QoE in the presence of ARQ. This improvement motivated the reflection of ARQ in QoE models in order to improve their prediction accuracy.

7.6 Future Work

Our study motivates future work in the following directions. First of all, in addition to a pure network layer centric measurement evaluation, future work should capture the video stream as decoded by the STB. This video stream will provide insights into visual impacts of packet losses and the ability of ARQ to improve QoE. Second of all, and most importantly, our findings should be adopted by future QoE models to account for loss recovery at client side.

8

HTTP Caching

Improving the performance and QoE of the Web is motivating research and engineering since more than a decade. Work in the mid to late 90s aimed at understanding why the Web was slow. This work particularly focused on the impact of web server performance and network conditions (mainly loss and delay) on web browsing performance (see e.g. [42]). At the same time, HTTP caching has been proposed to improve the performance of the Web by *i*) decreasing latency of web transfers and *ii*) masquerading network outages [43]. Besides performance improvements, caching is appealing to network operators as means for traffic reductions. Such traffic reductions lower the potential for network congestion that can have detrimental effects on QoE, as we have shown in Chapter 6. Given these benefits, HTTP caching offers a potential solution for improving the QoE of the web that we explore in this chapter. While buffering and caching are similar but unrelated concepts¹, both relate to (temporary) storage whose existence and size has implications on Internet performance and QoE.

Despite the potential benefits, HTTP caching has then been challenged by two factors. First of all, the advent of dynamic web sites and user generated content largely decrease the cachability of HTTP objects. Ager et al. [30] report only 10% of the user generated content to be cachable, while over 60% of the static content

⁰The content of this chapter is based on joint work with Gerhard Haßlinger and has been partially published in the International Teletraffic Congress [115] in 2010. While the publication focuses on an analytical evaluation of the contributed caching scheme, this chapter focuses on an extended empirical analysis and further discusses caching locations.

¹Buffering is the process of holding data during transfers, while caching stores data to accelerate subsequent requests.

like software downloads are cachable. The popularity of dynamic web pages and user generated content reduces the overall cachability down to 22%. Second of all, hosting advertisements, whose profitability depends on exposure metrics such as page impressions requires content providers to collect access statistics. Traditional caching in form of HTTP proxies or transparent caches, however, hinders content providers from collecting access statistics. This results in an intrinsic motivation of content providers to prevent caching. One way to reduce the cachability is by setting appropriate HTTP parameters such as the content expiration time or the cache control setting.

These challenges lead to the emergence of Content Distribution Networks (CDN) in the late 90s. CDNs operate a distributed network of caches that exclusively serves content provided by their customers. In contrast to traditional caching (e.g., proxies), requests to this content are redirected to caches ‘close’ to the originator. Such redirections optimize the overall web performance and promise performance optimizations that were once envisioned with traditional caching (e.g., proxies). Additionally, CDN caches are run by a single administrative domain that allows access statistics collection and distribution to its customers. These benefits made content distribution via CDNs popular. Their popularity manifests in traffic shares of more than 50% of the wireline broadband traffic [96, 204].

Despite the popularity of CDNs, caching in form of transparent caches is repeatedly discussed by ISPs as means to reduce traffic and to cope with traffic growth. In particular, the centralized structure of mobile networks that have only very few IP gateways hinders CDN deployment and suggests substantial benefits for operating caches (see e.g., [79, 81]).

Motivated by the potential for QoE improvements, this chapter re-visits HTTP caching by investigating a hit rate analysis of different caching schemes. Investigating the efficiency of caching schemes (object replacement strategies) for optimizing cache efficiency is a relevant research problem that is of importance for every cache, whether it is deployed in web browsers, ISP operated general purpose caches, or CDN caches. Significant traffic volumes of HTTP video [172, 80] and the widespread use of YouTube, lets us use YouTube object popularity traces for evaluating cache efficiency. For memoryless object requests, our simulation study shows that Least Recently Used (LRU) as traditional caching scheme can depart up to 15% from the optimum cache hit rate. This shows room for improvement and motivated us to propose a new caching scheme that maintains access statistics over a sliding window of L requests. With increasing size L , the scheme approaches the optimum cache hit rate that is obtained by caching the most popular items. Our evaluation shows that the proposed scheme offers higher cache hit rates than traditional LRU caches.

8.1 Cache Placement Scenarios

We start by reviewing possible locations for hosting caches in wireline and mobile broadband access networks. As the efficiency of caches in the outlined locations depends on many network and traffic dependent factors, this section gives an overview of possible cache locations rather than providing concrete placement strategies.

8.1.1 Caching Benefit

Caching can have clear *technical* benefits. It can reduce the latency when fetching objects as well as reducing link utilizations. The latter helps in mitigating network congestion that can significantly lower QoE (see Chapter 6). However, whether caching is deployed remains a *business* decision. This decision depends on a complex set of parameters, e.g., traffic pattern, content cachability, costs for maintaining caches, etc. This set of parameters is specific to particular use-cases and organizations. Thus, rather than completely characterising caching benefits, we next provide an *intuition* on some important parameters that drive caching decisions.

The efficiency of a cache storing a set $I = \{o_1, o_2, \dots, o_M\}$ objects depends on

a_k : the access frequencies of object o_k

s_k : the size of object o_k

$c_{Cache}(sk)$: the cost for caching object o_k . These costs involve the costs for operating the cache, including hardware (disc space, fault tolerance if needed, ...), software, operational costs (power, labor costs, ...), and maintenance.

$c_{L_j}^{BW}$: the cost for transferring objects over link L_j . These costs involve all costs for operating the link, including hardware (fiber, optical interfaces, line cards, router, ...), maintenance and operation (power, labor, ...) and traffic costs in case of peerings. Peering costs vary from being settlement free to being attached to transit costs that are negotiated on a contractual level between parties.

We remark that costs in the above framework can also be factors such as latency or QoE. While such factors are not monetary in the first place, they can translate to monetary costs. Examples include bad service quality / QoE that lowers the user satisfaction (see e.g., delay effects [20, 175]). Optimizing QoE can thus be a competitive advantage (see e.g., competitive advantages by improving service levels [58]).

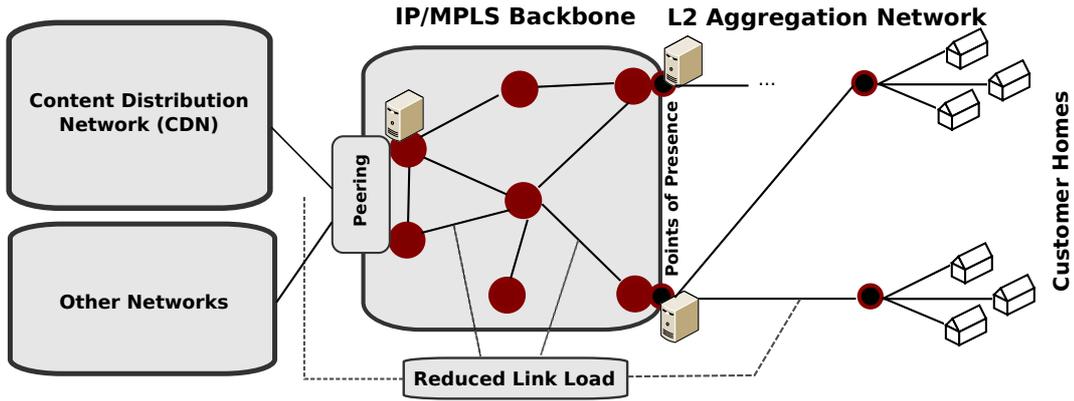


Figure 8.1: Possible cache placement locations in a broadband access network

Let L_1, L_2, \dots, L_S be links in the path that are bypassed by the use of caching. Caching object o_k is beneficial if

$$c_{Cache}(sk) < \sum_{j=1}^S a_k s_k c_{L_j}^{BW}. \quad (8.1)$$

Our estimation provides an intuition of costs attached to caching and possible benefits of caching. We remark that this approximation does not provide a complete framework for making a business case for caching. In general, the decision for operating caches and their concrete placement must be based on a detailed business case study. This case study must consider the architecture of the studied network and access pattern of cacheable content in the respective network. We next review cache placement scenarios for broadband and mobile access networks.

8.1.2 Broadband Access Networks

We show the architecture of a typical broadband access network in Figure 8.1. This architecture consists of a L2 aggregation network that aggregates traffic from the users' homes into points of presence (PoPs). PoPs are installed in different geographic locations (e.g., major cities) and aggregate traffic from regional aggregation networks.

Network operators have multiple options to install caches in their network. Possible locations include points of presence (PoPs), backbone routers, and at peering links. In general, the closer a cache is placed to the users' homes, the shorter the transmission path will be for cached objects and thus the lower the latency will be. On the other hand, such closely located caches serve a smaller user population with the potential consequence of lower efficiency.

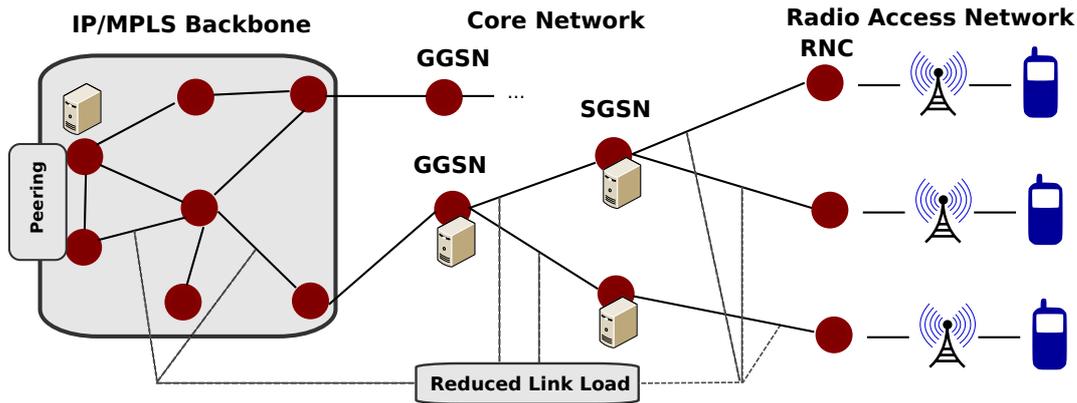


Figure 8.2: Possible cache placement locations in a mobile access network

The caching efficiency is not only affected by the size of the served population, but also by the nature of the generated traffic. In particular, regional access patterns can significantly differ from global access pattern. In the context of YouTube, this difference has been observed in [101] from the perspective of a campus network. In this network, the local popularity of YouTube videos differed significantly from the global popularity. Thus, only caching videos that are popular on a global scale would result in a low cache efficiency. To be efficient, caches should account for the different nature of the cached traffic.

As an alternative to operating general purpose caches, network operators can host dedicated cache servers operated by Content Delivery Networks (CDN). CDN caches denote a special type of caches as they only serve content hosted by the CDN. This form of caching can be beneficial, as recent studies report CDN based traffic to account for more than 50% of the wireline broadband access traffic [96, 204]. CDN caches additionally differ from traditional caches by the fact that the association of user to caches is performed by the CDN and not by the network operator. As this limits options for traffic engineering within networks, PaDIS [205] provides a collaborative scheme allowing network operators to influence the mapping of users to caches.

8.1.3 Mobile Access Networks

Mobile access networks differ from the previously discussed wireline access networks in their centralized network architecture. Based on Erman et al. [79], we next review the architecture of a typical 3G mobile access network in Figure 8.2. In this architecture, cellular devices connect to the Radio Network Controller (RNC) via the Radio Access Network (RAN). Traffic is then switched into the core network, and afterwards encapsulated in IP packets by the Serving GPRS Support Nodes

(SGSN) hosted in Regional Data Centers. The resulting traffic is finally aggregated by the Gateway GRPS Support Nodes (GGSN) hosted in National Data Centers.

The architecture of 3G networks often only consists of very few IP gateways (GGSN) that aggregate traffic originating from large geographic areas [267, 79]. This architecture challenges the ability of CDNs to bring content closer to the end-users and thus to reduce the amount of traffic in the network and serve it with lower latency. As a result, traditional general purpose caches show potential to optimize traffic flows in mobile networks. Based on measurements in a major 3G network, Erman et al. [79] reports hit rates of 33% for caches placed in National Data Centers. However, despite lower cache hit rates due to smaller populations, the study reports in-network caching in Regional Data Centers to be more beneficial due to network traffic reductions. Caching in Regional Data Centers translates to cost savings of 26.7%. But also the co-location of traditional caches or even CDN caches in National Data Centers is possible. We remark that GGSN located in National Data Centers provide the first IP hop and thus the first possible location for hosting CDN caches. Thus, caching in 3G networks is shown to have promising benefits.

8.2 Caching Schemes

As caches are designed to only store a limited number of M objects, object *replacement policies* decide which objects should be cached. We refer to such policies as caching schemes. We next review Most Popular Items and Least Recently Used (LRU). Motivated by potential optimizations of LRU, we propose a LFU based scheme in the last part of this section. This scheme offers higher cache hit rates than traditional LRU caches (see its evaluation in Section 8.3.4).

8.2.1 Notation

Let $I = \{o_1, o_2, \dots, o_N\}$ denote a set of N cachable objects, e.g., YouTube videos. The objects $o_k \in I$ are ordered in descending order according to their access probabilities $a_1 \geq a_2 \geq \dots \geq a_N$. We remark that I is often not fully known in a practically setting. In this case, the observed partial set $I_{measured} \subseteq I$ can be obtained by passive measurements or by the analysis of access statistics that are maintained by many web sites. Also, available measurement traces (see e.g., Section 8.3.1) provide means for fitting the access probabilities a_k to observed access frequencies.

For a set of users, we assume that requests to objects $o_k \in I$ are identical and independently distributed (i.i.d.). Thus, the requests are memoryless and do not depend on previous requests to the same object. We remark this assumption is violated by bursty arrivals. Such bursty arrivals can particularly occur in flash crowd scenarios, in which many users place requests a single content in a short

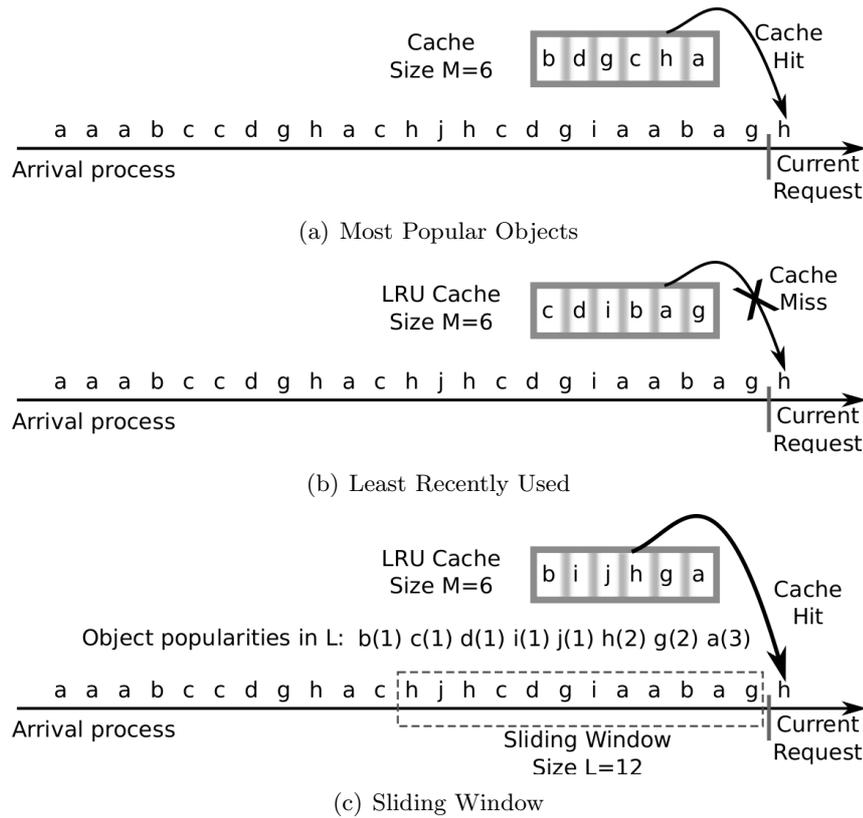


Figure 8.3: Cache schemes: object replacement strategies

amount of time. For the remainder of this chapter, we assume non-bursty arrivals and leave this special case for future work.

We further assume the cache C_M to have a limited storage for $M < N$ objects, independent of their size. We leave the hit rate analysis with varying object sizes for future work.

8.2.2 Most Popular Objects

For independent accesses, the maximum hit rate $R_{C_M}^{max}$ is achieved by pre-populating the cache with the M most popular objects. The resulting hit rate corresponds to

$$R_{C_M}^{max} = \sum_{i=1}^M a_i. \quad (8.2)$$

Figure 8.3(a) shows an example of this scheme. In this example, the arrival process is shown on the x axis with object h being the current request. A cache of size

$M = 6$ is pre-populated with the M most frequent objects in the depicted arrival process. As this assignment is done once, it is irrespective of the current arrivals. In this example, a request to the frequently occurring object h leads to a cache hit.

This caching strategy is optimal when the access pattern of a cache are constant or vary only slowly. However, often access patterns are quite variable over different time frames. Flash crowd effects (e.g., many users watching the Super Bowl) are one particular example of very dynamic access patterns. This variability in time requires the set of popular objects to be constantly adapted, which motivates us to discuss LRU as common adaptive scheme.

8.2.3 Least Recently Used

Least recently used (LRU) is a simple and typical cache replacement algorithm. The content of an LRU cache C_M of size M objects consists of the last M referenced objects. By ordering cached objects based on their access times, LRU is adaptive to dynamic object popularities. As a consequence, LRU keeps repeatedly accessed objects longer in the cache and has a preference for keeping the most requested objects.

As in the previous section, Figure 8.3(b) shows an example of an LRU cache. In this example, the arrival process of requests equals the one used in the previous example. The cache is populated with the $M = 6$ least recently used objects. However, despite its global popularity, a request to object h results in a cache miss, as it was not among the 6 least recently used objects.

8.2.4 Sliding Window

For non-bursty arrival pattern, LRU can depart from the optimum hit rate that is obtained by caching the most popular objects (see Section 8.3.3). To optimize the cache efficiency, we next propose a caching scheme that maintains access statistics over a sliding window containing the last L accessed objects. A cache of size $M \leq L$ is then populated with the M most frequent objects in L . The last step corresponds to a Least Frequently Used (LFU) object replacement policy. However, in contrast to classical LFU, access statistics are not maintained over cached objects, but instead over the sliding window. As the sliding window only needs to store object identifiers that are typically much smaller than the corresponding objects, the sliding window can be much larger and allows the computation of access statistics over extended time periods. We show an example of this scheme in Figure 8.3(c).

The behaviour and efficiency of this caching scheme is controlled by the sliding window size L . When only a short memory L of a few recent accesses is configured, significant random deviations in the cache hit rate are expected. In particular, $L = 0$ yields to a cache that is populated with arbitrary objects as no access statistics are

| YouTube Category | Description | # Videos | # Requests |
|------------------|----------------------|-----------|---------------|
| <i>Ent</i> | Entertainment | 1,687,506 | 3,697,618,703 |
| <i>Sci</i> | Science & Technology | 252,255 | 539,868,316 |

Table 8.1: Data set overview

available. The resulting hit rate corresponds to $R = M/N$. For $L = 1$ only the last access to object o_k is included in the access statistics. This results in a cache in which o_k is ranked first. The remaining space is then populated with $M - 1$ arbitrary objects. With increasing window size $L \gg M$, this scheme approaches the optimum hitrate that is obtained by caching the most popular items (see Section 8.2.2). However, with large values of L , the cache also becomes more static and less adaptive to changing object popularities.

In comparison, LRU maintains a window of M distinct objects. For $L = M$ the proposed scheme equals LRU when L distinct objects were requested. If the number of distinct objects in L is less than M , LRU yields higher hit rates.

The memory complexity of this scheme consists of storing M objects in a cache and L object identifiers in the sliding window. As the sliding window is used to compute object popularities, only identifiers need to be stored and not full objects. Identifiers can be SHA-1 hashes of URLs that are 160 bit in size and thus much smaller than objects. This property makes the sliding window more resource efficient than keeping larger LRU caches for achieving similar hit rates (see e.g., Table 8.3). For an analytical evaluation of this scheme, we point to our work in [115].

8.3 Evaluation of Cache Hit Rates

Video streams generate significant traffic shares that can lead to network congestion, e.g., in peering links. As we have shown in Chapter 6, network congestion negatively effects the QoE of video streams. We argue that caching can optimize the overall QoE, e.g., by reducing traffic load that can cause congestion. This provides an alternative / addition to mitigating congestion by bandwidth reduction discussed in Chapter 4. In this section, we evaluate the efficiency of different caching algorithms based on a measured YouTube video popularity distribution. Caching of YouTube content is possible, as passive measurements suggest YouTube video to exhibit a high cacheability [101, 30].

8.3.1 Data Set

We base our analysis on two YouTube video popularity traces provided by Cha et al. [54]. These traces capture access to YouTube videos placed in two categories

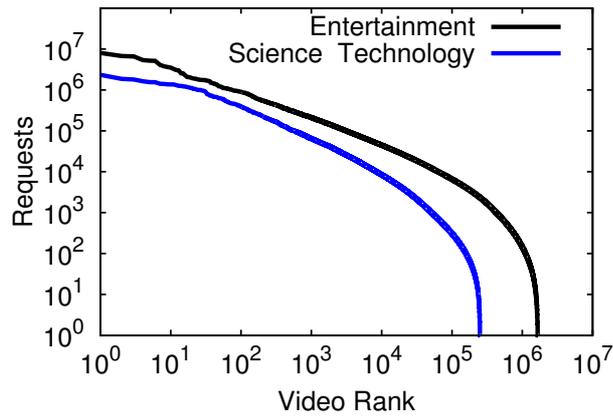


Figure 8.4: YouTube object popularities

Ent (Entertainment) and *Sci* (Science & Technology) as shown in Table 8.1. The traces provide aggregated per-video information covering 3,6 billion requests to 1,6 million distinct videos in the Entertainment category, and of 0,5 billion requests to 0,2 million distinct videos in the Science & Technology category, respectively.

We next illustrate the distribution of video popularities in both traces. Figure 8.4 shows the video rank vs. the number of requests on a log-log scale for both video categories. The figure shows a linear decline of object popularity in the log-log space. This decline is typical for Zipf distributed popularities that are often reported for web object popularities [79, 51]. The tail of both distributions show an exponential cut-off that doesn't follow the Zipf distribution. We refer to [54] for a more detailed analysis of access patterns in YouTube.

As request popularities in both categories follow the same distribution, we focus on the largest trace (*i.e.*, *Ent*) in our caching analysis. In terms of our caching algorithms (see Section 8.2), the trace represents the set I containing $|I| = 1.6\text{M}$ video objects $o_k \in I$. We set their access frequencies a_k to the empirically measured frequencies obtained from the trace. Further, we select a subset $I_{50k} \subseteq I$ of 50,000 randomly selected videos to simulate smaller objects sets. Smaller sets can occur in categories that contain less videos or at web sites that host less content than YouTube as one of the major video platforms.

8.3.2 Simulation Methodology

We implemented the caching schemes presented in Section 8.2 in a simulator. The simulator reads object probabilities a_k from the input traces and generates requests to objects $o_k \in I$. The objects are randomly chosen based on their object probabilities a_k . For each cache configuration (L, M) , we generate 10M requests to the

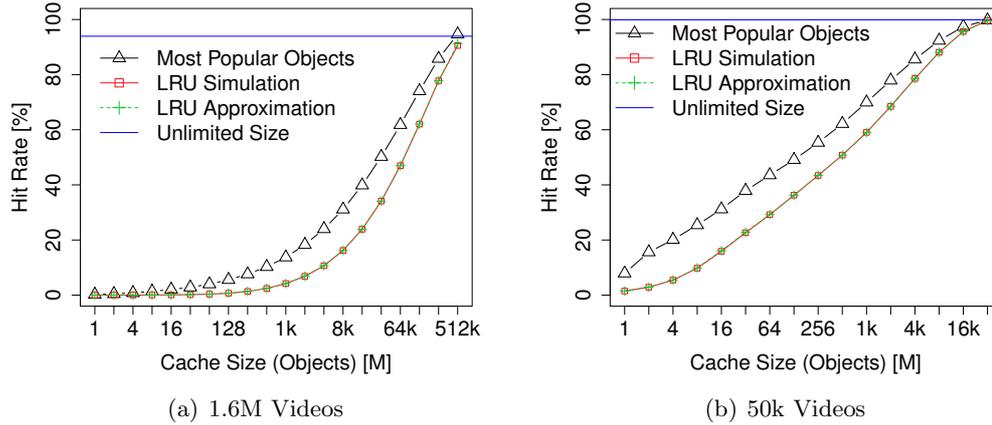


Figure 8.5: LRU Simulation Results

objects and count the hit rates after an initial warm-up phase of 1M requests. Once completed, the simulator computes the hit rate over the remaining 9M requests. As the smaller item set only consists of 50k objects, we scale down the simulation and generate 500k requests after an initial warm-up phase of 100k requests.

In the absence of information on how object popularities in the used traces change over time, we assume the Zipf object popularity distribution depicted in Figure 8.4 to be static over the simulation period. In other words, the most popular object remains to be the most popular object. In practice, *individual* object popularities are changing over time. This change can, however, be rather slow, e.g., as the ranking of popular content in file sharing communities has been reported to be stable over weeks [159]. As a consequence, we believe simulations with static object popularity distributions not to be unrealistic and leave simulations with dynamic object popularities for future work.

8.3.3 Least Recently Used

We first focus on evaluating the hit rate of LRU caches subject to YouTube video popularity pattern. Recall example LRU deployments that include proxy servers (see e.g., [24]), Android and iOS based smartphones [211], CDNs (see e.g., [2, 152]), and industrial networking devices [80]. LRU thus is the most widespread caching scheme and therefore studied as standard caching scheme. Figure 8.5(a) shows the simulation results for LRU caches of different size given the complete YouTube trace containing 1.6M video objects. In addition to simulation results, the figure shows the hit rate according to an LRU approximation by Dan and Towsley [68] (line overlapping with simulation result), and the hit rates of an optimum cache of size M pre-populated with the M most popular objects. Figure 8.5(b) shows the same analysis but for the smaller objects set I_{50k} . To keep the size of the cache M less

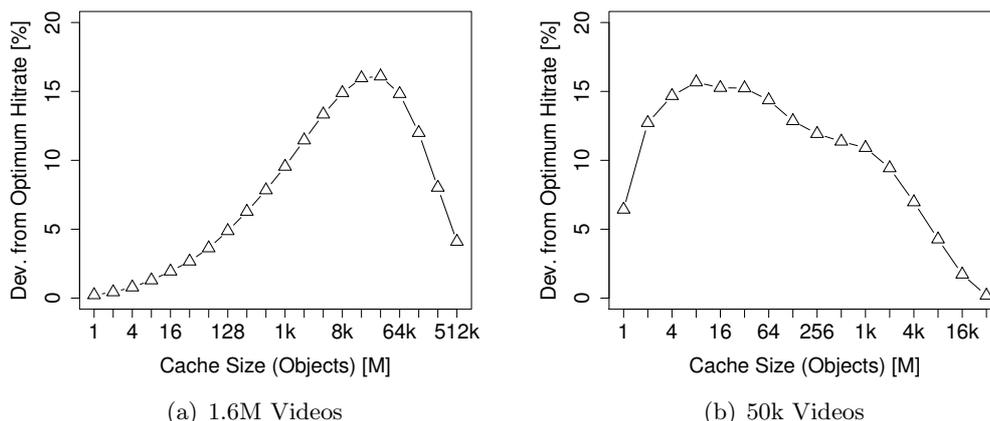


Figure 8.6: LRU Cache: hit rate deviation from optimum cache

than the total number of distinct videos in the trace ($M \leq N$), we limited M to the number of distinct videos.

We first observe an almost exact fit of the simulation results to the theoretical LRU approximation (both lines in the plot overlap). This finding indicates the proper functioning of our simulator. Additionally, we performed several simulation runs. As the deviation in hit rates between these runs is small, we keep the figures simple by only showing results for a single run.

We additionally simulated a cache with unlimited capacity to cache every requested object. As the hit rate does not depend on the cache size M , it is shown as horizontal line in Figure 8.5. This caching scheme denotes an upper bound for the achievable hit rate as it does not need object replacement. Note that this cache does not achieve 100% cache hit rate, as objects that are requested for the first time after the warmup phase result in cache misses. As this cache miss probability is higher for a larger number of objects, the hit rates in Figure 8.5a) is lower than in Figure 8.5b).

To compare the hit rate of an LRU cache to an optimum cache, we show the deviation of LRU from the optimum cache in Figure 8.6. LRU remains up to 15% below the optimum hit rate obtained by caching the most popular objects. This result shows potential for optimization.

8.3.4 Sliding Window

We next focus on evaluating the hit rate of the Sliding Window caching scheme that was proposed in Section 8.2.4. As in the last section, Figure 8.7 shows the hit rates obtained by simulation for both YouTube traces. Both figures compare LRU hit rates to Sliding Window caches of different sliding window sizes $L = M$, and $L = 32M$. The figures show that for $L = M$, Sliding Window caches approximate

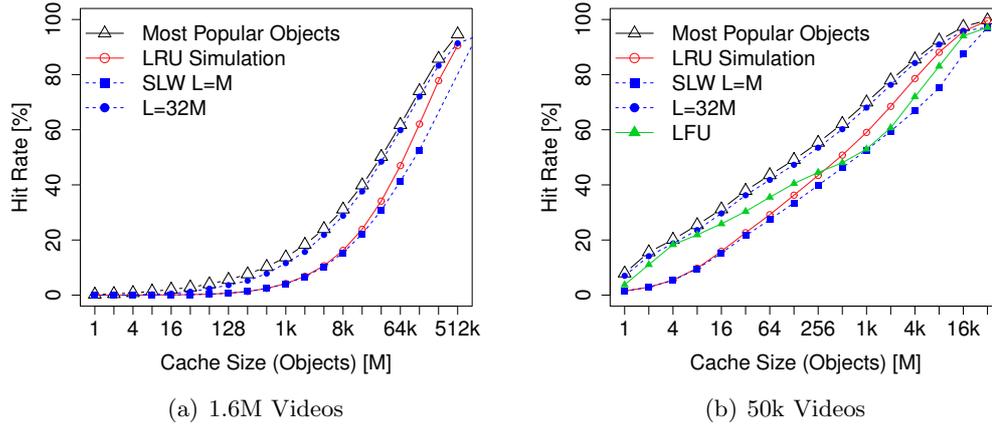


Figure 8.7: Sliding Window simulation results

LRU for small cache sizes and depart from LRU for larger caches. This effect is explained by the number of distinct videos in the sliding window. Given the Zipf distributed object popularities shown in Figure 8.4, the number of distinct objects in a sliding window of size L typically is $\ll L$ due to popular and often requested objects. However, for configurations with large $L = 32M$, Sliding Window caches approach the optimum hitrate and outperform LRU. We repeated the simulation several times and found similar behavior.

We additionally simulated a classical Least Frequently Used (LFU) cache and show its results in Figure 8.7. LFU yields higher hit rates for smaller caches but is outperformed by LRU for larger caches. Further, Sliding Window caches with $L = 32M$ offer higher hitrates than LFU and LRU, and approach the optimum hit rate. Recall that Sliding Window caches use LFU as object replacement strategy. In contrast to traditional LFU caches, access statistics are computed over object references in the sliding window rather than over objects in cache. With increasing L , this allows for longer history periods that help to approach the optimum hitrate as shown in the figure.

We take a closer look into this effect by showing the deviation of the simulated LRU and Sliding Window caches from the optimum hit rate in Figure 8.8. The figure also shows deviations for additional configurations $L = 2M$, $L = 4M$, and $L = 8M$. It shows the tendency to approach the optimum hit rate with increasing sliding window size L . For $L = M$, the figure shows significant random deviations for larger caches in which the hit rate even increases (e.g. 4k). Due to Zipf distributed object popularities, the number of distinct objects in L is $k \ll M$. This limits the ability to compute access statistics to $k \ll M$ and causes significant random deviations for the remaining $M - k$ slots in the cache.

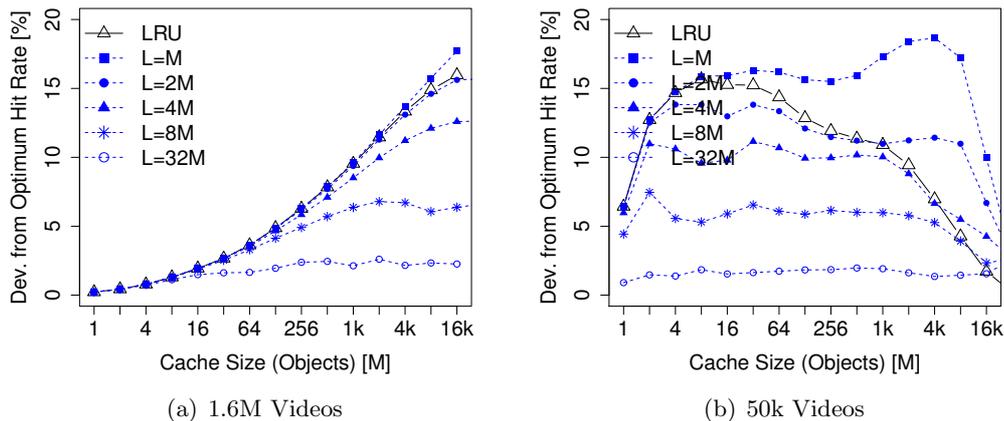


Figure 8.8: Sliding Window Cache: hit rate deviation from optimum cache

| Resolution | Size in MB |
|------------|------------|
| hd720 | 59 |
| large | 13.6 |
| medium | 10.2 |
| small | 6 |

Table 8.2: Size of a 152 secs long YouTube video in different resolutions

To summarize, we find that Sliding Window caches offer higher hit rates than LRU when the sliding window size is $L \gg M$. They approach the optimum cache hit rate with increasing sliding window size.

8.3.5 Traffic Reductions

We are next interested in estimating traffic reductions that can be achieved with the evaluated caching schemes. As the traces only contain video lengths but not the corresponding video object sizes, we computed the mean (166 secs) and median (100secs) video length. We then downloaded a random video whose length of 152secs is close to the mean. The resulting video sizes per resolution setting are shown in Table 8.2. For an upper bound on achievable traffic reduction, we focus on HD720 as the highest video resolution.

We remark that videos in the analyzed YouTube traces vary in length, which hinders us from accurately estimating traffic reductions. Given these limitations, we next *approximate* traffic reductions based on the values provided in Table 8.2. We leave a detailed analysis of traffic reductions due to caching for future work.

Table 8.3 shows hit rates and traffic reductions for three caching schemes: most popular objects, LRU, and a Sliding Window of size $L = 32M$. The table focuses on

caching videos in HD720p resolution. As used in the simulation, the table assumes 400,000 requests. We show hit rates for the I_{50k} object set. For each cache size M , we report the disk space required to store M objects, each 59MB in size. For the Sliding Window, the table additionally shows the memory required to store a sliding window of size $L = 32M$ hash references, each are SHA-1 hashes of 160 bit length.

The table reports LRU to deviate from an optimal cache by one to three TB of additional traffic. The Sliding Window scheme, however, closely approaches the optimum as shown in the last section. Compared to LRU, traffic reductions in the order of one to three TB are possible at the expense additionally storing a sliding window. As the sliding window is only populated with hash references that allows for counting object popularities, it is rather small in size. Compared to operating larger LRU caches, the sliding window scheme achieves higher hit rates while requiring less disk space.

| Objects | Cache Size | | Optimum Cache | | LRU | | Sliding Window | |
|---------|------------|----------|-------------------|----------|-------------------|-------------|----------------|-------------------|
| | Bytes | Hit Rate | Traffic Reduction | Hit Rate | Traffic Reduction | Window Size | Hit Rate | Traffic Reduction |
| 1 | 59 Mb | 7.9% | 1.8 TB | 1.4% | 342.5 GB | 0.6 kB | 7.3% | 1.6 TB |
| 32 | 1.8 GB | 37.8% | 8.7 TB | 22.6% | 5.2 TB | 20 kB | 36.8% | 8.4 TB |
| 256 | 14.75 GB | 55.3% | 12.7 TB | 43.4% | 10 TB | 160 kB | 53.8% | 12.4 TB |
| 1024 | 59 GB | 69.9% | 16.1 TB | 59% | 13.6 TB | 640 kB | 68.8% | 15.8 TB |
| 4096 | 236 GB | 85.5% | 19.7 TB | 78.5% | 18.1 TB | 2.5 MB | 84.5% | 19.4 TB |
| 8192 | 472 GB | 92.3% | 21.2 TB | 88.1% | 20.3 TB | 5 MB | 91.5% | 21 TB |
| 16384 | 944 GB | 97.3% | 22.4 TB | 95.6% | 22 TB | 10 MB | 96.3% | 22.2 TB |
| 32768 | 1.8 TB | 99.7% | 22.9 TB | 99.5% | 22.9 TB | 20 MB | 97.1% | 22.3 TB |

Table 8.3: Traffic reductions for three caching schemes: optimal cache, LRU, and Sliding Window of size $L = 32M$.

8.4 Related Work

HTTP caching has been proposed in the 90s to reduce web server and link load, and thus to optimize the performance of the web [43]. Traditional HTTP caches have then been challenged by the emergence of Content Distribution Networks (CDN) in the late 90s. Despite their popularity, traditional caching is repeatedly discussed by ISPs as means to reduce traffic. In particular, the centralized structure of mobile networks with only very few IP gateways hinders CDN deployment and offers substantial benefits for operating traditional caches (see e.g., [79, 81]).

The cachability of HTTP objects has been evaluated in several studies. Ager et al. [30] analyzes cachability from the perspective of an ISP. Based on passive measurements of residential access traffic, they find only 10% of the user generated content to be cachable, while over 60% of the static content like software downloads are cachable. Similarly, Erman et al. [80] find 30% of all HTTP traffic in wireline access and backbone networks to be cachable. By evaluating passive measurements captured in a campus network, Gill et al. [101] report YouTube videos to be cachable. Video streaming in 3G networks was evaluated by Erman et al. [81]. Their study finds that 92% of all the requested video object chunks were cachable and 24% of all requested bytes could be served from caches. These works show that HTTP caching can still yield substantial traffic reductions, in particular for video traffic. In contrast to work on the *cachability* of HTTP objects, our study is motivated by substantial traffic savings achieved by caching, in particular for video content and focuses on a cache *hit rate evaluation*. This evaluation is based on empirically measured YouTube video popularities and several caching schemes.

Caching schemes (cache replacement strategies) were a fundamental research problem several decades ago for paging in operating systems [149, 95]. With the advent of HTTP caching, caching schemes also became a popular research problem in networking that has created a rich body of literature. Least Recently Used ranks objects according to their access times and represents the most widespread caching scheme. Many refinements have been proposed in the literature, including the object size based LRU-MIN [27] and LRU-SIZE [236], the cost-to-size based object ranking (SLRU) [31], and the size-adjusted and popularity-aware LRU-SP [55]. An approximative model for LRU hit rates has been proposed by Dan and Towsley [68]. Other studied caching schemes include First In First Out [68] and Least Frequently Used (LFU) [230]. Other schemes combine several schemes, e.g., Hyper-G [26] combines LFU and LRU such that objects are first ranked by LFU and then by LRU. Another class of caches adopts a Time To Live (TTL) based policy to remove objects whose TTL expired. DNS represents one popular TTL based system. TTL based caches have recently become popular in the modelling community, as they offer a general policy that is able to recover LRU, FIFO, and random behavior and additionally makes the analytic analysis of tandem networks feasible [92]. For an overview of existing caching schemes we refer to several surveys [40, 203, 253, 26, 69].

Despite the rich body of proposed caching schemes, LRU remains the most widely used scheme. Concretely, LRU is the default scheme in the open-source Squid proxy [24], it is used by HTTP libraries on iOS and Android Smartphones [211], and is used by Akamai as major CDN as suggested in [2, 152]. This widespread deployment of LRU lets us focus on evaluating its performance on the analyzed YouTube traces.

Beyond classical cache replacement strategies, we propose an LFU based scheme that computes access statistics over a sliding window and approaches the optimum hit rate for large sliding window sizes. Based on access frequencies computed over the sliding window, it replaces cached objects in a LFU manner. LFU has been reported to offer higher hit rates than LRU [40]. However, LRU has been criticised for weighting unpopular objects too high, whereas LFU weights old objects too high [123]. The proposed scheme gives control over the number of objects used for the ranking by adjusting the sliding window size L as history over the last L requests.

8.5 Discussion

This chapter reviewed caching as means to optimize web QoE. From the end-user perspective, caching offers promising potential for increasing web QoE by reducing latencies and link loads, which in turn helps avoiding congestion. Latency improvements can immediately improve web QoE as web enterprises have shown that users respond to speed [175, 237, 170]. One QoE quantification of latency is provided by the ITU G.1030 standard (see Section 3.3.4). Further, sustainable congestion can detrimentally impact QoE as shown in Chapter 6. Network operators on the other hand show interest in caching as means to cope with traffic growth and the required network capacity expansion by reducing link load (see e.g., [115]).

These aspects summarize clear *technical* benefits supporting caching. To revisit our starting point, caching is, however, challenged by dynamic content with low cachability (e.g., user generated content) [30] and by the difficulty for content providers to maintain access statistics. On the other hand, recall that static content, e.g., images and videos, that contributes large traffic volumes [172, 80] shows good cachability [30, 101]. This cacheability drives the deployment of CDN cache networks that already account for significant traffic shares in broadband traffic. This widespread usage of CDNs shows the willingness of content providers to pay for optimized content delivery.

Network operators can benefit from caching by either hosting CDN caches or by operating general purpose caches (e.g., proxies or transparent caches). Hosting general purpose caches is challenging as it typically does not involve cooperation with content providers and CDNs (e.g., to maintain access statistics for marketing and billing purposes). Which option remains beneficial from a *business* perspective thus has to

be answered for each operator individually. To support this decision, this chapter provided a brief intuition of caching benefits and reviewed cache placement scenarios in wireline and mobile access networks. In particular the centralized structure of mobile access networks hinders CDN deployment and shows substantial potential for traffic savings in the order of 20 to 30% (see e.g., [79, 81]).

Motivated by this potential, this chapter re-visited caching and presented a cache hit rate analysis of different caching schemes. Investigating the efficiency of caching schemes (object replacement strategies) for optimizing cache efficiency is a relevant research problem that is of importance for every cache, whether it be deployed in web browsers, ISP operated general purpose caches, or CDN caches. Significant traffic volumes of HTTP video [172, 80] and the widespread use of YouTube, were the motivating factors for using YouTube object popularity traces for evaluating cache efficiency in this chapter. In particular, HTTP video traffic is estimated to grow at rates of 80% per year [80]. For memoryless object requests, our simulation study showed that LRU as traditional caching scheme can depart up to 15% from the optimum cache hit rate. Depending on the use case, one might desire higher cache efficiencies, e.g., in memory constrained settings such as in-memory caching or the usage of flash drives such as SSDs for fast cache access, or in memory constrained smartphones. In other scenarios, larger disk space can allow for larger caches with higher hit rates. However, as object sizes are also growing, cache efficiency remains a relevant aspect. This room for improvements motivated us to propose a LFU based caching scheme that maintains access statistics over a sliding window of L requests. With increasing size L , the scheme approaches the optimum cache hit rate that is obtained by caching the most popular items. Our evaluation shows that the proposed scheme offers higher cache hit rates than traditional LRU and LFU caches.

8.6 Future Work

Future work should extend the presented study in the following way. Our analysis is based on the assumption of identical and independently distributed (i.i.d.) requests to objects. This assumption does not hold in cases of bursty arrivals, e.g., flash crowd effects. Also, we assumed object popularities to be static over the measurement period. Future work should therefore evaluate a more broad set of access distributions with dynamic object popularities. In addition, traffic reductions and cache hit rates are subject to non-constant object sizes. As object sizes were not reported in the data set used in this study, we leave a detailed analysis of traffic reductions with varying object sizes for future work. This study should then not only consider traffic reductions, but should account for diurnal effects and time dependent access pattern. Such pattern will influence link utilizations that should additionally be evaluated.

More broadly, future work should assess impacts of different caching schemes in different caching scenarios (e.g., CDN caches vs. proxy caches vs. browser caches in desktop computers or mobile phones).

9

Excursion: E-Mail Spam & Address Harvesting in the Internet

With an estimate of 2,4 billion users worldwide [243], e-mail is the most widespread communication service in the Internet. In this service, the presence of unsolicited bulk e-mail (spam), which has exceeded the volume of legitimate e-mail, remains a costly economic problem that largely degrades the QoE of e-mail communication. Notwithstanding existing counteracting measures, spamming campaigns advertising products are profitable even when the amount of purchases being made is small relative to the amount of spam [141]. The large market penetration of e-mail and the apparent success of spamming campaigns motivates the understanding of spamming trends and their economics, which may provide insights into more efficient counteracting measures that finally improve the end-user QoE.

In this excursion, we broaden our view by looking at a major communication system and spam as arguably a major QoE determinant. We go beyond the buffering perspective outlined in this thesis and aim on *optimizing end-user QoE by mitigating spam*. E-Mail QoE is currently an unexplored area in QoE research. By this excursion, we take a novel but conceivably extreme and controversial perspective on QoE that can stimulate a discussion on QoE impact factors on e-mail as major communication system.

While defining a concrete QoE model for e-mail is out of scope of this thesis, we assume spam to impact the utility of e-mail and to further contribute an annoyance

⁰The content of this chapter is joint work with Thomas Graf and Florin Ciucu and has been published at the ACM Internet Measurement Conference [120] in 2012.

| Site | Type | Country | Start of Rnd IDs | Issued Rnd IDs (% spammed) | Issued MTO Rnd IDs (% spammed) | Issued Name IDs (% spammed) | End |
|------|---------------------|---------|------------------|----------------------------|--------------------------------|-----------------------------|------------|
| A | Private blog | DE | 2009-05-16 | 791,890 (0.23%) | 144,769 (0.45%) | 211,851 (0.12%) | 2010-11-29 |
| B | Gaming web site | DE | 2009-05-16 | 2,807,925 (0.06%) | 469,804 (0.19%) | 929,147 (0.03%) | 2012-08-24 |
| C | Private web site | DE | 2009-05-16 | 21,558 (0.53%) | 3,890 (1.54%) | 5,938 (0.12%) | 2011-03-28 |
| D | Mail archive | DE | 2009-05-16 | 5,191,288 (1.75%) | 917,836 (3.20%) | 1,518,105 (0.68%) | 2012-08-24 |
| E | Private web page | DE | 2009-05-17 | 1,097 (0.00%) | 197 (0.00%) | 320 (0.00%) | 2012-08-17 |
| F | Private web page | DE | 2009-05-16 | 400,490 (0.54%) | 70,424 (1.47%) | 118,481 (0.09%) | 2011-10-30 |
| G | Spamtap page | DE | 2010-01-14 | 998132 (0.29%) | 166,408 (0.54%) | 332,694 (1.07%) | 2012-08-24 |
| H | Research group | DE | 2010-01-24 | 7,582,332 (0.07%) | 1,372,051 (0.17%) | 2,094,329 (0.04%) | 2012-08-24 |
| I | Fake email provider | US | 2010-07-09 | 34,500 (0.19%) | 5,750 (0.26%) | 11,500 (0.03%) | 2011-05-16 |

Table 9.1: Data set overview

factor that also impacts e-mail QoE. These assumptions are motivated by prior work showing spam to lower productivity [235], to largely causes costs for users and enterprises, e.g., for providing bandwidth or anti-spam software [190, 235], and to impede economic growth [248]. In an attempt to create the first e-mail QoE model, future work should extend these early works and further explore factors influencing e-mail QoE that are to be combined in a model describing e-mail QoE (see Section 9.6).

Further, despite major improvements in filtering accuracy, spam filters are prone to false positives that can reduce QoE by classifying desired e-mail as spam. Thus, rather than focusing on spam filtering, we aim at exploring the origins of spam. Our focus on address harvesting mitigates spam by studying methods to prevent addresses from being harvested in the first place. Therefore, we argue that understanding spam origins and consequently spam prevention methods can lend itself to optimize QoE.

To explore the origins of the spamming process, we conduct a large scale study involving addresses harvested from public web pages. Concretely, to identify address harvesting crawlers, we have embedded more than 23 million unique spamtrap addresses in more than 3 million visits to web pages over the course of more than three years, starting in May of 2009. 0.5% of the embedded addresses received a total of 620,000 spam e-mails. The uniqueness property of the embedded spamtrap addresses enables the mapping between the crawling activity to the spamming process.

9.1 Methodology & Datasets

To study the properties of the address harvesting process of harvesters using web crawlers, we use a methodology relying on issuing unique spamtrap e-mail addresses via the web. As the addresses are uniquely generated for each page request, their usage can be directly mapped to a specific page request once the first spam is received. The generated addresses are embedded into nine low-profile web pages of various types (gaming, private web pages, research group, etc., see Table 9.1) and popularities. This methodology is implemented in web sites by including a dynamic script that generates unique e-mail addresses for each page request and

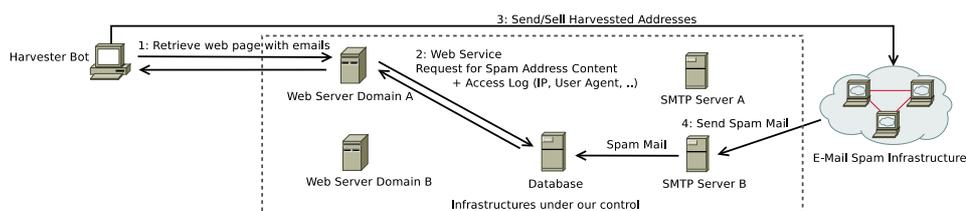


Figure 9.1: Measurement methodology

logs information about the visitors. The resulting distributed platform to advertise our spamtrap addresses and to receive spam is illustrated in Figure 9.1.

Webmasters are typically confronted with the dilemma of choosing a method for displaying e-mail addresses on the web: Should e-mail addresses be presented in a user-friendly (high QoE) or obfuscated way to prevent spam (low QoE)? Which presentation method is the most robust against address harvesters? To shed light on this dilemma, the information included in the web pages of our study consists of six different spamtrap addresses, each being displayed with one of the following presentation and obfuscation techniques: *i*) a *mailto:* link (**MTO**), *ii*) non-linked, plain-text address (**TXT**), *iii*) e-mail obfuscated in the form of *user [at] domain [dot] tld* (**OBF**), *iv*) obfuscated using Javascript code (**JS**), *v*) included in a hidden data field of a web form (**FRM**), and *vi*) plain-text address inside an HTML comment (**CMT**). All the above described addresses consist of random strings of 10 characters each (*RND IDs*, e.g., "jdi4gj8bzx"). We use random strings as they are sufficiently hard to guess. Starting in January 2010, and in addition to random strings, we issue realistic looking addresses containing random combinations of first and last names generated from phone book records (*Name IDs*, e.g., "john.doe"). As the total number of possible *firstname* \times *lastname* combinations is much smaller than the total number of possible random IDs, we only issue name IDs using the *MTO* embedding method, to avoid running out of addresses. Compared to random strings, the assumption is that realistic looking addresses are harder to identify as spamtrap addresses, but are also easier to guess. Table 9.1 shows the total number of embedded IDs per web page, as well as the respective measurement periods. Note that the number of random IDs correlates with the number of monitored page requests for each web site.

E-mail addresses are advertised by appending different domains and TLDs. Our e-mail domains are handled by several mail exchange servers located in different networks. We consider any e-mail sent to trap addresses as spam.

As our web pages cover a variety of different genres and popularities, this selection is arguably representative. By monitoring a relatively small number of web pages concentrated in Germany, the conclusions of this study are conceivably biased. However, this bias creates the opportunity to look at a focussed set of web pages and study locality in the harvesting process.

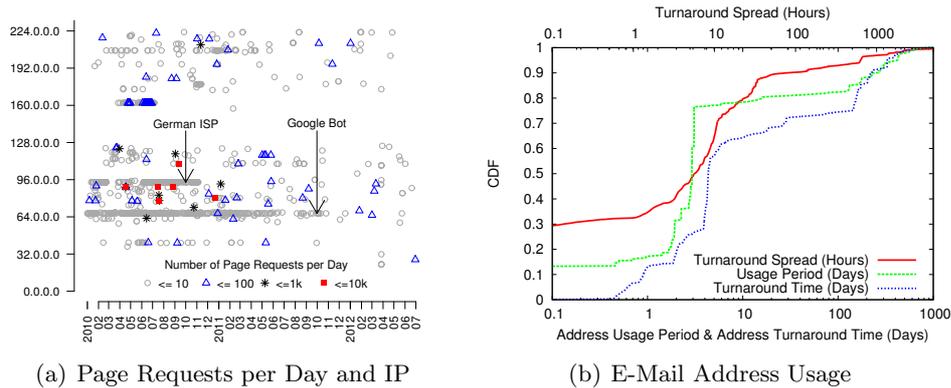


Figure 9.2: Bot visit and spam properties

9.2 Harvest and Spam Activities

This section presents the main properties of address harvesting bots. We present statistics on page requests made by bots, geolocation of bots, and the usage of our spamtrap addresses.

9.2.1 Network Level Properties

In total, we classified 1251 hosts as harvesters and obtained DNS records for 90% of the hosts. For the remaining 10%, no DNS record could be obtained from the authoritative DNS servers. Inspecting a random subset of hosts led to mostly access networks. To our surprise, we classified 20% of the hosts as search engines whose requests originated from legitimate address spaces associated to Google, Microsoft, and Yahoo. We discuss this issue further in Section 9.3.4.

To study how requests by harvesting bots are spread over time, address space, and volume, Figure 9.2(a) shows a volume classification for the requests per day and IP. For most of the bots, only a few page requests resulting in spam can be observed (maximum 9871 requests per IP and day). A few regions in the IP address space show activity over multiple months, visible as horizontal bars. We found DSL customers by a German ISP to be the most dominant ones in March and July to August of 2010. However, the Google bot showed the longest time stability.

The figure also shows several heavy-hitters; six hosts retrieved around 10,000 pages—corresponding to 80,000 e-mail IDs—each on a single day. Manually inspecting these hosts revealed that most of the IP addresses belong to a single provider in Romania. We found 24 distinct IPs originating from this network, none of them having a DNS record. Page requests by these IPs span over almost the entire monitoring period

and are responsible for a major fraction of the received spam. We observed requests to five of our web sites, of which 99% belong to web site D (mail archive).

We were further interested in whether harvesting machines are primarily hosted by infected machines located in residential or business access lines, or by operating dedicated servers. For this classification, we *i*) firstly apply a reverse DNS lookup to obtain host names and *ii*) secondly look for specific text patterns in the obtained host names. We classify hosts as DSL or Cable hosts if their host name contains key words such as “dsl”, “customer”, “dialin”, etc. According to our classification heuristic, 73% of the IPs belong to ADSL or Cable access providers. This shows that harvester bots are still primarily run in residential access networks.

To study further properties of the hosts running bot software, we collected statistics about open TCP/UDP ports by port scans since mid 2011. To reduce the traffic caused by port scans and to focus our scans on harvesters, we limited our port scans to hosts blacklisted by Project HoneyPot (note that we can only mark harvesters in retrospect after the first spam arrival). Only 13 hosts that we scanned harvested e-mail addresses from our web sites. Six of the scanned hosts had port 3389 open (typically used for remote control of Windows systems), four port 22 (remote control of Unix systems), and five port 80 (HTTP).

By solely looking at the number of distinct IPs per country, the bias in our data set towards web sites in Germany is reflected in the geolocation of harvesting machines: 60.6% of all bot IPs are located in Germany. However, are the German harvester bots also responsible for most of the spam volume? By looking at the total spam volume caused per harvesting location leads to a different distribution; harvester bots in Romania and Bulgaria caused 72% of the received spam. The German bots that made up for 60% of all the IPs were only responsible for 10% of the spam.

9.2.2 E-Mail Address Usage

What happens to e-mail addresses after they were harvested? We investigated this aspect by focusing on the usage of harvested addresses. Concretely, we denote the time between the address being harvested and their first usage as the *turnaround time* and show its distribution in Figure 9.2(b). 50% of the addresses received the first spam e-mail within four days after being harvested. The slowest observed turnaround was 1068 days, whereas the fastest was 64 seconds. Focusing the analysis only on addresses advertised to search engines led to slower turnaround times (not shown in the figure); 50% of the addresses were spammed within 11 days after a visit by a search engine bot. We found the fastest turnaround to be one day, while the slowest was 611 days.

As we embedded multiple addresses in one page, we were interested whether they were also simultaneously used for the first time. Therefore, we selected all RND addresses which received more than one spam e-mail (78%) and grouped them by

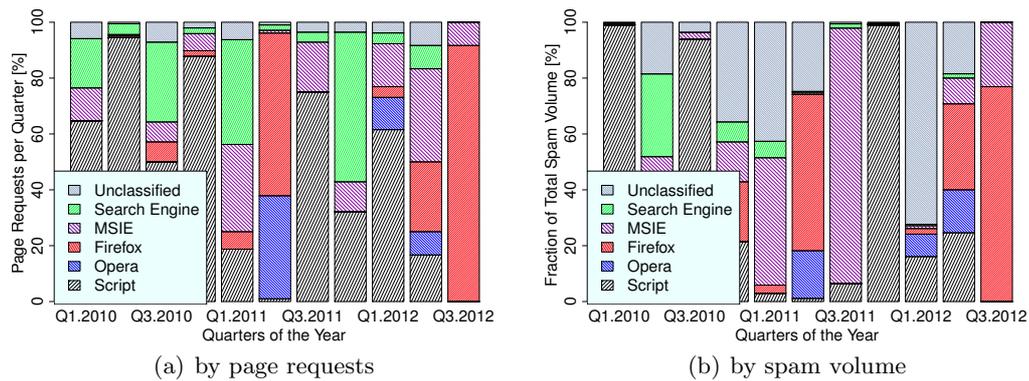


Figure 9.3: User Agent strings of harvesting bots

page requests. We denote the *spread* in turnaround times as the range (max-min) of turnaround times for the addresses embedded in one request. A low spread indicates that all addresses in one request were firstly spammed within the same period. The distribution of the spread in hours is shown in Figure 9.2(b) (note the upper axis). 80% of the pages show a spread of less than a day (99% for search engines), and 27% of 0 seconds (94% for search engines) meaning that all extracted addresses simultaneously received their first spam. This finding suggests that spam to our spamtrap addresses was mainly sent in batches.

We also computed the amount of time that our addresses receive spam, denoted here as the *usage period*. 11% (16% for the search engines) of all addresses which received at least two spam e-mails were used for less than a second, 17% (40%) for less than a day, and 78% (51%) for less than a week. The longest observed usage of an e-mail address was 1068 days (749 days). We mention that our monitoring period spanned over 1202 days.

9.3 Spam Mitigation

We next focus on aspects that can be used for spam mitigation, and can thus lead to improvements in the overall QoE. Concretely, we study fingerprints of bots, the robustness of methods used to display e-mail addresses on the web, the efficiency of blacklisting, the usage of anonymity services, and the role of search engines.

9.3.1 Fingerprinting: User Agent Strings

In Figure 9.3 we show the usage of user agent strings submitted by harvesters bots in the HTTP header of the page request to our web sites. Figure 9.3(a) shows the

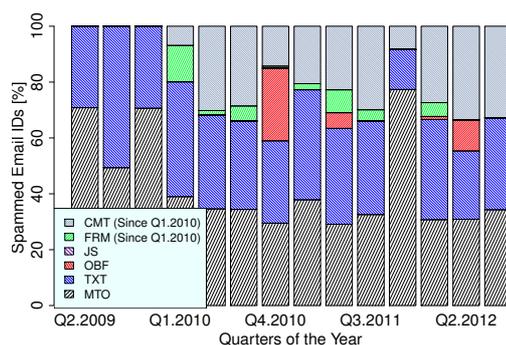


Figure 9.4: Spammed e-mail IDs by presentation method

use of user agents on a per-request basis. Note that the variability is modulated by heavy-hitters as shown in Figure 9.2(a); thus a per-IP classification shifts the popularity of user agents.

Note the variation over time visible in Figure 9.3(a) and 9.3(b) (also in Figure 9.4). One inference from this observation is the existence of only a few parties that harvest addresses from our sites. Depending on their activity, they can strongly skew a given quarter’s statistics. Also, differences in Figure 9.3(a) and 9.3(b) suggest that the addresses harvested by different parties are not homogeneously used and caused different spam volumes.

We observed that 19% of the classified hosts as harvesters submitted a user agent string mimicking those of major search engines. By resolving IP addresses to AS numbers, we find that only 5% of the hosts using the Google bot user agent do not originate from the Google AS and are thus mimicking the Google bot. These 5% of the hosts are located in various ISPs and hosting sites (including Amazon EC2) located in seven different countries. Checking the *whois* records for each IP revealed various providers that cannot be associated directly to Google. 95% are indeed legitimate requests from the true Google AS. We did not observe a case of faked user agents for Yahoo’s Slurp and Microsoft’s Bing bots.

Seven years after the study of Prince [209], we find the Java user agent (e.g. “Java/1.6.0_17”), classified in our figures as “Script” and reported by [209], still to be present. Visits by this user agent span over the entire data set. We find this user agent to be used by 3% of the hosts classified as harvester. However, these hosts account for 88% of the page requests leading to spam, whereas spamtrap addresses harvested by these hosts account for 55% of the total spam, indicating that our data is skewed by one type of harvesting. In particular, the majority of the hosts located in Romania (cf. Section 9.2.1) supplied the Java user agent string. These visits caused 99.9% of the spam volume that can be traced back to harvester bots in Romania. We found only a single host submitting six requests using “Mozilla/2.0

(compatible; NEWT ActiveX; Win32)” as user agent. This finding indicates that the usage of some harvesting software is fairly stable.

9.3.2 Address Presentation Method Robustness

One aspect concerning webmasters is how to display e-mail addresses on the web to prevent spam: in a user-friendly (high QoE) or in an obfuscated way (lower QoE)? To address this issue, we study the robustness of presentation techniques by displaying our spamtrap addresses using a set of different presentation and obfuscation methods. For each spamtrap address which spam, we show the relative share of spammed addresses for the used presentation methods in Figure 9.4.

As expected, a significant portion of spam was received by addresses presented in easy to parse plain text or as *mailto:* link. While some of the plain text obfuscated addresses (OBF) were harvested, none of the addresses presented using Javascript code received any spam. Concerning addresses advertised to search engine bots, the majority of the spammed addresses were presented using MTO (60.7%) and TXT (38.4%). Bots using the Java user agent only parsed addresses presented using MTO and TXT. These findings suggest that simple obfuscation methods, in particular Javascript, are still quite efficient to protecting addresses from being harvested.

9.3.3 Efficiency of Blacklisting & Usage of Anonymity Services

We query the IP based spam blacklist provided by Project HoneyPot for each page request to our monitored sites at the time of the visit. Blacklist data has been collected over a period of 13 months since July 2011 and aims to evaluate the efficiency of blacklisting for blocking harvester bots. During this period, we received visits by 318 hosts that were classified as harvesting spam bots. 26% of the visiting hosts were marked by Project HoneyPot as harvester.

In addition, we were interested if harvesters use anonymity services—such as Tor—to hide their identity. While the default configuration of Tor exit nodes blocks traffic to port 25—used to send spam—access to port 80—used to retrieve e-mail addresses from the web—is not prohibited by default. This could make Tor more attractive to harvesters than to spammers. To check if requests originated from the Tor network, we queried the list of Tor exit nodes when a page was requested. Tor usage statistics were collected over a period of five months starting in 2012. In this period, only 0.03% of the total requests to our web sites originated from the Tor network. However, we did not receive any request using Tor that was classified as harvesting activity.

We note that the short evaluation time conceivably biases these observations, but suggests that harvesters do not make an effort to conceal their identity.

9.3.4 Harvesting: Role of Search Engines

We now address the role of search engines in the context of address harvesting. To our surprise, we received spam to spamtrap addresses advertised only to major search engine bots, i.e., Google, Microsoft, and Yahoo. In particular, 0.5% of the spamtrap addresses delivered *only* to search engines received 0.2% of the total spam. We define visits by search engines as requests made by crawlers that originate from the Google, Microsoft, and Yahoo ASes. All the visits originating from those ASes used the proper user agent of the respective search engine bot. Concretely, 13% of the hosts classified as harvester originated from the Google AS, 3.7% from the Microsoft AS, and 1.2% from the Yahoo AS. We observe this behavior on all sites over the entire measurement period (cf. Figure 9.2(a) for the Google bot).

While the impact of the harvesting techniques to the overall spam volume and the number of harvested addresses is rather small, that very existence is a surprising result that has not been previously reported. It suggests that harvesters use search engines as a proxy to either *i*) hide their own identity or *ii*) optimize the harvesting process itself. As harvesters did not try to hide their identities by either using anonymity services or by masquerading as legitimate browsers by sending common user agent strings, option *ii*) seems more likely. In fact, we were able to find harvesting software that offers the functionality of querying search engines. Concretely, the advertisement for ECrawl v2.63 [189] states: “Access to the Google cache (VERY fast harvesting)”. The Fast Email Harvester 1.2 “collector supports all major search engines, such as Google, Yahoo, MSN” [75]. This finding suggests that web site operators should not advertise e-mail addresses to search engine bots. It also calls for a further systematic investigation.

9.4 Related Work

Many studies address the properties of spam e-mails, traffic, and campaigns [104, 272, 197, 217], infrastructures for spam dissemination (e.g., botnets) [108, 265, 139, 197, 272, 217, 271, 130], and detection and classification methods [269, 34, 108, 114, 122, 139, 271, 154, 148]. For instance, it has been shown that spam and non-spam traffic have significantly different properties which can be used for spam classification [104]. Much fewer studies address the origins of the spamming process, e.g., concerning *address harvesting*, which remains the primary means for spammers to obtain new target addresses. Addresses can be harvested in multiple ways, e.g., from public web pages by using crawlers [209] or by malicious software locally running on compromised machines [157]. Investigating the harvesting processes is particularly relevant as it leads to new insights about spammers, according to studies revealing the social network of spammers [266] or a rather superficial effort to conceal identity [209].

The method of identifying harvesting bots by issuing dynamically created addresses that are unique to each page request has been used for spam prevention and the identification of harvesters [223, 209, 228]. The first attempts in understanding the behavior of harvesters have been undertaken by Prince et al. [209] and Schryen [223] in 2005. Based on 2500 spam e-mails, Schryen [223] investigates whether the top level domain of an e-mail address is relevant for spammers and finds that .com addresses attract more spam. A more systematic study of address harvesting was done by Prince et al. [209] by using a distributed platform using 5000 participants to advertise spamtrap addresses and receive spam over a period of six months (Project Honey Pot). Aspects of address harvesting were revisited by Shue et al. [228] in 2009. Their study is based on 96 spam e-mails and studies the geolocation of harvesters, strength of presentation methods, turnaround times, and the aggressiveness of harvester bots expressed by the frequency of page visits.

As trends in the world of spam and malware are changing fast, this study presents an up-to-date view on address harvesting and content spamming. To the best of our knowledge, we are the first to present a large-scale data set spanning over more than three years that combines aspects of harvesting and comment spamming. In contrast to [223] and [228], our spam body consists of 620,000 spam e-mails and is larger by magnitudes. While we confirm previous findings, we also study new aspects such as *i)* the efficiency of blacklisting, *ii)* the usage of the Tor anonymity service, *iii)* host-level properties of bots, and *iv)* the usage of search engines as proxies to hide the identity of harvesters.

9.5 Discussion

This excursion aimed at broadening the view of QoE by discussing spam as arguably a major QoE determinant in e-mail. E-Mail QoE is currently an unexplored area in QoE research. By this excursion, we took a novel but conceivably extreme and controversial perspective on QoE that could stimulate a discussion on QoE impact factors of e-mail as major communication system. Rather than focusing on spam filtering, we aimed at exploring the origins of spam. Despite major improvements in filtering accuracy, spam filters are prone to false positives that can reduce QoE by classifying desired e-mail as spam. Our focus on address harvesting mitigates spam by studying methods to prevent addresses from being harvested in the first place.

We have presented a longitudinal study of address harvesting that is based on a large-scale data set and that gives an up-to-date view on spam origins. We show that some aspects of harvesting are fairly stable over time, e.g., the existence of a certain user agent that has been observed for years, and the poor performance of harvesting software in breaking obfuscation methods. One interpretation of our results suggest that only a few harvesting parties are active, each causing different spam volumes. We also find that new aspects arise in the harvesting process, such as

the emerging trend in the usage of legitimate search engines as proxies for address harvesting. Other observations point to the decline of harvesting activity on our sites and the existence of only a small set of hosts being responsible for a major fraction of the received spam.

In order to optimize the QoE of e-mail, our findings reveal several guidelines for webmasters to mitigate spam, e.g., *i*) to continue using obfuscation methods for displaying e-mail addresses on the web, e.g., by using Javascript code, *ii*) to restrict embedding e-mail addresses in web sites sent to legitimate browsers, and in particular not to search engine bots, *iii*) to rely on blacklists, e.g., provided by Project Honey Pot, to limit the likelihood of address harvesting.

9.6 Future Work

We see two main directions for future work arising from this excursion.

The first direction follows on the presented work and focuses on getting a better understanding of e-mail address harvesting. Example questions concern the number of active harvesters and their relationship to spammers and botmasters. Are there only a few major active harvesters as suggested by our work? Are spammers and harvesters the same parties? Are botnet resources rented for harvesting as they are rented for spamming? Preliminary work in this direction complemented the presented harvester study with a classification of the spamming botnets and the advertised spam campaigns. Initial results show that different botnets are used for different campaigns, which suggests that the harvested addresses are sold on the market.

The second direction aims at a first exploration of e-mail QoE. This specifically concerns studying factors influencing QoE. Following the QoE definition from Section 3.1, we see the following influence factors.

- E-mail currently provides an unreliable communication service that doesn't always inform the sender about unsuccessful e-mail delivery. We therefore argue the **reliability** of e-mail to impact its *utility*. Undelivered e-mail without further notification is arguably perceived as *annoying*.
- Further, the **timeliness of delivery** denotes a second factor concerning e-mail delivery. Based on prior experience in using e-mail, users will arguably have *expectation* regarding delivery performance. Failures in meeting these expectations will impact QoE.
- The presence of unsolicited e-mail—i.e., **spam** as studied in this excursion—contributes *annoyance* when using e-mail. Unsolicited e-mail becomes particularly annoying when users receive immediate notifications of new e-mail, e.g.,

as often used on mobile phones and PCs. Spam further impacts *user efficiency* by requiring automatic or manual filtering.

Future work should deepen this understanding and further explore other communication services (e.g., chat services, in particular mobile chat services that are gaining popularity). A detailed investigation of QoE impacts not only touches on currently unexplored ground, but can also provide useful insights for *i*) service monitoring and operation and *ii*) the design of future communication services.

10

Conclusion and Outlook

10.1 Summary

The Internet has become an essential part of the lives of millions of people and an invaluable asset to businesses. As an emerging trend, data storage and processing is shifting to the Cloud (e.g., Google Apps, or Cloud gaming), making users more and more dependent on the network to perform their daily activities. Despite the crucial importance of Internet services, they remain susceptible to bad service quality. A particular factor influencing service quality is buffering at various layers.

This thesis assessed the impact of buffering on Quality of Experience (QoE). QoE is an active research area aiming to quantify the users' perception of applications and Internet services. This is challenging since the users' perception is subjective. We tackled this challenge using a multi-disciplinary approach that combined QoE and networking research to take a cross-layer perspective on network and application buffering.

Network buffering occurs in hosts, switches, and routers throughout the Internet. It impacts network performance by contributing delays, jitter, and packet losses. We illustrated that packet losses, mainly caused by congestion and overflowing buffers, have detrimental effects on video QoE. Motivated by this observation, we discussed Scalable Video Coding (SVC) as means for QoE management to optimize video QoE in phases of congestion. SVC allows for seamless adaption to varying network conditions. In this setting, we studied bandwidth reductions achieved by either *i*) video resolution reductions, *ii*) image quality reductions, or *iii*) frame rate reductions. We evaluated the impact of these changes on common full-reference QoE

metrics. This evaluation of bandwidth adaptation mechanisms showed that resolution changes yield better overall QoE scores and higher bandwidth savings than frame rate reductions. Further, we evaluated QoE impacts of model based packet loss generators used in QoE studies. We showed that model choice impacts QoE and proposed a new fitting technique that is optimized for replicating aspects relevant to video QoE.

The size of network buffers influences network performance by controlling the level of introduced delay, jitter, and packet loss. Surprisingly, even after decades of research and operational experience, ‘proper’ buffer dimensioning remains an unresolved and controversially debated topic. Most recently, this debate has focused on the negative effects of large buffers. This led to proposed Internet engineering changes, despite the absence of sufficient empirical evidence. As an understanding of buffering effects is crucial before altering important engineering aspects, we broadly studied the impact of buffer sizing on QoE in an extensive study involving relevant user applications (e.g., voice, video, and web browsing), real hardware, and realistic workload. This study showed that the dominant factor for the QoE is the level of competing network workload. That is, workloads in which the competing flows keep the queue at the bottleneck link filled (e.g., via many short-lived and therefore not congestion controlled flows) have much larger impact on QoE than buffer size. The study additionally showed that exacerbated (bloated) buffers have a significant effect on QoE metrics. Reasonable buffer sizes that follow standard sizing guidelines, however, have a significant effect on QoS, but impact QoE metrics only marginally. This led us to conclude that limiting congestion, e.g., via QoS mechanisms or over-provisioning, may actually yield more immediate improvements in QoE than efforts to reduce buffering.

Application buffering is used to compensate for performance variations, e.g., originating from network buffering. One example is the proprietary retransmission scheme deployed in a major IPTV system. This thesis revealed insights in the functioning of this proprietary scheme. We showed that the resend mechanism deployed by Set-Top-Boxes in a major IPTV network is based on a simple buffer-based resend scheme that offers drastic QoE improvements for low loss rates. When QoE metrics are used by ISPs for QoE monitoring inside the network, they do not account for error recovery mechanisms at the edge and thus are prone to QoE mispredictions. By revealing insights into the resend mechanism used by a major IPTV network, we paved the way for improved QoE metrics accounting for error recovery mechanism. To optimize Web QoE, we performed a hit rate analysis of caching schemes by focusing on YouTube video popularities. The thesis contributed a new caching scheme that offers higher cache hit rates than traditional Least Recently Used caches.

Finally, we broadened the view of QoE by discussing spam as major QoE determinant in e-mail. A large-scale study conducted over 3.5 years revealed insights into address harvesting as the origin of spam and proposes mechanisms for spam mitigation that can help to improve e-mail QoE. The study showed that harvester can often be

identified by their HTTP user agent string and that simple address obfuscation methods sufficiently protect e-mail addresses from being harvested.

10.2 Future Directions

We see several directions for future work arising from this thesis. On a short-term perspective, we see subjective tests as a natural extension of our work. On the long-term perspective, we see for instance meta CDNs driving the competition in the content delivery market. This competition will further drive the need for advanced QoE metrics assessing the user experience of HTTP content delivery.

Short-term perspective

Subjective Tests. This thesis focused on evaluating the impact of buffering on objective QoE *metrics*. Such QoE metrics are designed for approximating certain aspects of user perception. Relying on QoE metrics was a key design decision that enabled the automatic exploration of a large state space. This exploration highlighted regions that impact QoE and regions that do not alter QoE estimates, e.g., the insensitivity of QoE to router buffer sizes when buffers are not filled vs. detrimental effects of large buffers that are sustainably filled. QoE models on the other hand face the challenge of approximating the complex human perception. As such approximations are naturally subject to inaccuracies, the clear next step is to verify our evaluations by using *subjective tests*. After using objective QoE metrics to reduce the complex state space to a few interesting anchor conditions, subjective tests now become feasible.

HTTP Video QoE. Besides the classical web, HTTP is increasingly used for video delivery in the Internet. While classical QoE works focus on UDP based video delivery, that has fundamentally different error characteristics, recent works started to explore the area of HTTP based video streaming. Works such as [171, 161, 128] provide a promising first step in this direction. In terms of QoE models for web browsing, ITU recommendation G.1030 [11] is still regarded as a standard. Recent advances in web QoE, e.g., [238], challenge the classical model and motivate advanced QoE metrics for the web.

Long-term perspective

Long-Term QoE Integration. This thesis joined a rather technical debate on QoE impact factors that omitted economic aspects and business impacts. The impact of buffering on phone call quality was one example question we tackled. While

investigating such QoE impacts is clearly an important problem, it should be extended to also cover business impacts. Example questions are: After how many impaired calls would a customer tolerate before canceling his contract? How is the long-time perception integrated over time; do people remember bad QoE if one in 1000 calls has a low call quality? Little is known about how individual call quality contributes to the perception of an entire service (see Section 3.4 for a discussion of first steps in this direction). Such questions are subject to ongoing debates that has recently started in the QoE community. While the answers are not yet fully known, we can identify the CDN market as an area in which network performance aspects impacting QoE will lead to business impacts.

QoE Aware CDN Selection. Delivering content with high QoE is the core business of a CDN. Customers pay CDNs efficiently serving their content with low latencies. Contractual periods span longer time frames and CDNs are not switched on a per request basis. With the advent of meta CDNs, however, the need for delivering good QoE gets higher than ever. If content is hosted by multiple CDNs, meta CDNs select the best CDN for each request based on objective performance criteria. Such an additional objective assessment will put pressure on CDNs and can intensify the relevance of factors impacting network performance, e.g., buffering. Additionally, meta CDNs need QoE aware metrics that support the CDN assessment and selection process.

QoE in the Wild. Quality of Experience studies involve testing a—typically—*small* set of subjects in controlled laboratory settings. While such experiments are clearly necessary for some studies, they are inherently limited in size and do not address service usage in the wild. Future work should therefore go beyond small-scale Quality of Experience studies; Passive Internet measurements combined with applied user experience models enable the unique opportunity to carry out *large-scale* QoE studies assessing QoE in the wild.

This challenge consists in gathering large service usage data sets to infer the QoE of millions of Internet users. Nowadays, operators collect large data sets capturing the interaction of millions of Internet users. In addition, users make more often use of cloud services to outsource data and programs to the Internet, which creates large amounts of usage data. The availability of such data can conceivably lead to a fundamentally new perspective on Internet QoE and thus to a deeper understanding of the Internet's architectural challenges.

Acknowledgements

Theorem. *There exists a non-negative buffer $P \subset W = \{i : i \text{ is human}\}$ of people with $|P| = n$, $n \in \mathbb{N} \wedge n > -1 \cdot e^{\pi i}$ to acknowledge.*

Proof. \$ buffer flush end credits

Best Advisor Award

Anja

Thanks for making this a unique experience, far beyond my expectations

Best Committee Award

Alex, Anja, Paul, Sebastian

Thanks for your effort in working through this thesis and serving on my committee

Best Math Award

Gerhard Hasslinger

Thanks for always impressing me with new models and showing me that everything can be expressed in math terms.

Community Award

FG INET + all of my students.

Thanks to all of my colleagues for the wonderful time in the group, the good collaborations, the joyful distractions, and all the fun we had during teaching periods when managing and giving classes, designing and grading exams, etc. Special thanks to Arne + Carlo + Gregor (e.g., for showing me that buying loudspeakers can last for months), Ben (e.g., for showing me how to get car manufacturers interested in Internet measurements), Bernhard (e.g., for countless hours of photography nerding, biking, and unsuccessfully convincing me not to use Emacs), Dan (Segor fun), Fabian (introducing to the “Grober Unfug” stamp that simplifies exam grading), Gregor (Mate and the numerous boardgames sessions), Enric + Florin + Ingmar (showing me what a lousy squash player I am), Nadi (inventing rocket mesh), Thomas (thanks for showing me that Dortmund is more than a city :). Thanks for countless other joyful moments!

Also thanks to all of my grad and undergrad students for your wonderful time and input. Thanks for having put up with me as advisor.

Best Host Award

Paul

Thanks for all of your support and for providing me a pleasurable stay in Madison, the lovely city in the mid-west with the Rathaus and the Essen Haus

Long Distance Award

Mike Blodgett

Thanks for helping me countless times and maintaining the WAIL testbed that I’m remotely using since almost two years now.

Best Dinner Award

Dave Plonka

Thanks for the awesome dishes!

Buffer Underrun Award

Enric

Thanks for the fantastic work when everything seemed so bloated

Research Methodology Award

Tommy Zinner

Thanks for the “pen and paper technique” and the extensive QoE discussions

Big 5 Award

Florin

Thanks for chasing the big 5 in South Africa and many thanks for endless support in project acquisition, student advising, paper writing, and many other things.

Proof Reading Award

Arne, Anja, Ben, Daniel, Nadi, Sebastian

Special Thanks

Katrin

Best Reader Award

You

Thanks for reading up to this point :)

Last but not least, we thank all the anonymous spammers and harvesters for making our spam study possible.

No data was harmed in the production of this thesis.

List of Figures

| | | |
|------|---|----|
| 1.1 | Thesis overview | 6 |
| 2.1 | Logical structure of the Internet | 10 |
| 2.2 | Buffer architecture of Cisco's 12000 backbone-grade router series | 19 |
| 3.1 | A dual-system view of QoE | 25 |
| 4.1 | Measurement setup | 36 |
| 4.2 | Impact of packet loss on quality scores | 37 |
| 4.3 | Tradeoff between packet loss and content quality | 39 |
| 4.4 | Quality impact of video resolution | 42 |
| 4.5 | Quality impact of different frame rates | 43 |
| 4.6 | Video clip blue sky | 49 |
| 4.7 | Video clip crowd run | 49 |
| 4.8 | Video clip park joy | 49 |
| 5.1 | Fit of the estimated Markov models | 58 |
| 5.2 | Visual video quality properties for the freezing concealment method. | 60 |
| 5.3 | Visual video quality properties for the slicing concealment method. | 61 |
| 5.4 | Average number of freezing events per video | 62 |
| 6.1 | Occurrence of queueing in the wild | 67 |
| 6.2 | End-user geolocations | 68 |
| 6.3 | Testbeds used in the study | 72 |
| 6.4 | Mean queuing delay (ms) for the access networks. Delays likely to degrade QoE colored in red. | 77 |
| 6.5 | Link utilization in an asymmetric access link | 78 |
| 6.6 | MOS scales used in this study | 80 |
| 6.7 | Median Mean Opinion Scores (MOS) for voice calls with different buffer sizes and workloads | 80 |
| 6.8 | Median VoIP MOS scores for backbone networks | 82 |
| 6.9 | Median MOS (color) and SSIM (text) for HD and SD RTP video streams in the access network | 85 |
| 6.10 | Median MOS (color) and SSIM (text) for HD and SD RTP transmission in the backbone | 86 |
| 6.11 | WebQoE: Median MOS (color) and page loading times (text) | 89 |

| | | |
|------|---|-----|
| 6.12 | Backbone WebQoE | 90 |
| 6.13 | YouTube quality indicators for HD720 resolution | 94 |
| 7.1 | Measurement setup | 103 |
| 7.2 | De-Jitter buffer model | 104 |
| 7.3 | Delay between requests for the uniform loss dataset | 107 |
| 7.4 | Number of request and retransmission packets for the uniform loss dataset with traces of length 130 sec | 107 |
| 7.5 | Correction efficiency for uniform packet loss | 109 |
| 7.6 | Network and server load for uniform packet loss | 111 |
| 7.7 | Network and server load impact for uniform packet loss and normally distributed jitter with a mean=10ms | 111 |
| 7.8 | QoE with and without MSTV ARQ | 112 |
| 8.1 | Possible cache placement locations in a broadband access network | 118 |
| 8.2 | Possible cache placement locations in a mobile access network | 119 |
| 8.3 | Cache schemes: object replacement strategies | 121 |
| 8.4 | YouTube object popularities | 124 |
| 8.5 | LRU Simulation Results | 125 |
| 8.6 | LRU Cache: hit rate deviation from optimum cache | 126 |
| 8.7 | Sliding Window simulation results | 127 |
| 8.8 | Sliding Window Cache: hit rate deviation from optimum cache | 128 |
| 9.1 | Measurement methodology | 137 |
| 9.2 | Bot visit and spam properties | 138 |
| 9.3 | User Agent strings of harvesting bots | 140 |
| 9.4 | Spammed e-mail IDs by presentation method | 141 |

List of Tables

| | | |
|-----|--|-----|
| 2.1 | Queuing delays resulting from different buffer sizes assuming 1500 bytes packets | 16 |
| 3.1 | Overview of QoE metrics | 26 |
| 3.2 | Mapping of video quality metrics to MOS scores [201, 256] | 28 |
| 4.1 | Properties of reference sequences | 36 |
| 6.1 | Workload configurations for both testbeds, where the flow interarrival time distributions are specific to the access and backbone testbed; exp-a has a mean of 2 sec and exp-b a mean of 1 sec. The “weibull” file size distribution is defined as Weibull (shape=0.35, scale=10039), resulting in a mean flow size of 50 KB. The number of concurrent flows at the bottleneck link is shown by their mean. Link utilization and loss measures are obtained for buffers configured according to the BDP. See text for further details. | 73 |
| 6.2 | Buffer size configurations and corresponding maximum queuing delays for both testbeds (packet size = 1500 bytes) | 75 |
| 6.3 | Selected YouTube video clips | 93 |
| 8.1 | Data set overview | 123 |
| 8.2 | Size of a 152 secs long YouTube video in different resolutions | 128 |
| 8.3 | Traffic reductions for three caching schemes: optimal cache, LRU, and Sliding Window of size $L = 32M$ | 130 |
| 9.1 | Data set overview | 136 |

Bibliography

- [1] <http://info.iet.unipi.it/~luigi/dummynet/>.
- [2] Akamai technical faq's. http://research.microsoft.com/en-us/um/people/ratul/akamai/technical_faq.doc.
- [3] Bufferbloat.net. <http://www.bufferbloat.net/>.
- [4] Nist net home page. <http://snad.ncsl.nist.gov/nistnet/>.
- [5] Tcpdump. <http://www.tcpdump.org/>.
- [6] tshark. <http://www.wireshark.org/docs/man-pages/tshark.html>.
- [7] ITU-T recommendation P.800: Methods for objective and subjective assessment of quality, 1996.
- [8] ITU-T recommendation G.1010: End-user multimedia QoS categories, 2001.
- [9] ITU-T Recommendation 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, 2001.
- [10] Cisco Systems: How to read the output of the show controller frfab — tofab queue commands on a Cisco 12000 series Internet router. http://www.cisco.com/en/US/products/hw/routers/ps167/products_tech_note09186a008009431f.shtml, 2005.
- [11] ITU-T Recommendation G.1030: estimating end-to-end performance in IP networks for data applications, 2005.
- [12] ITU-T recommendation P.862 Annex A: Reference implementations and conformance testing for ITU-T Recs P.862, P.862.1 and P.862.2, 2005.
- [13] Extended RTP profile for real-time transport control protocol (RTCP)-based feedback (RTP/AVPF). RFC 4585, 2006.
- [14] ITU-T recommendation E.800: Quality of telecommunication services: concepts, models, objectives and dependability planning. Terms and definitions related to the quality of telecommunication services, Sept. 2008.

- [15] ITU-T recommendation P.910: Subjective video quality assessment methods for multimedia applications, 2008.
- [16] Bufferbloat: What's wrong with the Internet? *ACM Queue* 9, 12 (Dec. 2011), 10:10–10:20.
- [17] ITU-T recommendation G.107: The E-model: a computational model for use in transmission planning, 2011.
- [18] ITU-T recommendation P.863: Perceptual objective listening quality assessment, Jan. 2011.
- [19] Cisco visual networking index: Forecast and methodology, 2011-2016, May 2012.
- [20] Heise Online: Online-Gamer klagen über Störung im Telekom-Netz. <http://www.heise.de/newsticker/meldung/Online-Gamer-klagen-ueber-Stoerung-im-Telekom-Netz-1762506.html>, Dec. 2012.
- [21] ITU-T recommendation P.1201.2: Parametric non-intrusive assessment of audiovisual media streaming quality - higher resolution application area, Oct. 2012.
- [22] Qualinet white paper on definitions of Quality of Experience. European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Patrick Le Callet, Sebastian Möller and Andrew Perkis, eds., Lausanne, Switzerland, Version 1.1, June 2012.
- [23] Mawi traffic archive: Sample point f, traffic trace 2013-03-20. <http://mawi.wide.ad.jp/mawi/samplepoint-F/2013/201303201400.html>, 2013.
- [24] Squid configuration directive `cache_replacement_policy`. http://www.squid-cache.org/Doc/config/cache_replacement_policy/, 2013.
- [25] ABBOUD, O., ZINNER, T., PUSSEP, K., OECHSNER, S., STEINMETZ, R., AND TRAN-GIA, P. A QoE-aware P2P streaming system using scalable video coding. In *IEEE P2P* (2010).
- [26] ABRAMS, M., STANDRIDGE, C. R., ABDULLA, G., FOX, E. A., AND WILLIAMS, S. Removal policies in network caches for world-wide web documents. In *ACM SIGCOMM* (1996).
- [27] ABRAMS, M., STANDRIDGE, C. R., ABDULLA, G., WILLIAMS, S., AND FOX, E. A. Caching proxies: Limitations and potentials. In *International World Wide Web Conference* (1995).
- [28] ADHIKARI, V. K., GUO, Y., HAO, F., VARVELLO, M., HILT, V., STEINER, M., AND ZHANG, Z.-L. Unreeling Netflix: Understanding and improving multi-CDN movie delivery. In *IEEE INFOCOM* (2012).

-
- [29] AGER, B., CHATZIS, N., FELDMANN, A., SARRAR, N., UHLIG, S., AND WILLINGER, W. Anatomy of a large European IXP. In *ACM SIGCOMM* (2012).
- [30] AGER, B., SCHNEIDER, F., KIM, J., AND FELDMANN, A. Revisiting cacheability in times of user generated content. In *IEEE Global Internet Symposium* (2010).
- [31] AGGARWAL, C., WOLF, J. L., AND YU, P. S. Caching on the world wide web. *IEEE Trans. on Knowl. and Data Eng.* 11, 1 (Jan. 1999), 94–107.
- [32] AKHSHABI, S., BEGEN, A. C., AND DOVROLIS, C. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP. In *ACM conference on Multimedia systems* (2011).
- [33] ALLMAN, M. Comments on bufferbloat. *ACM CCR* 43, 1 (Jan. 2013), 31–37.
- [34] ANDROUTSOPOULOS, I., KOUTSIAS, J., CHANDRINOS, K. V., PALIOURAS, G., AND SPYROPOULOS, C. D. An evaluation of Naive Bayesian anti-spam filtering. In *Workshop on Machine Learning in the New Information Age* (2000).
- [35] APPENZELLER, G., KESLASSY, I., AND MCKEOWN, N. Sizing router buffers. In *ACM SIGCOMM* (2004), pp. 281–292.
- [36] ARGYROPOULOS, S., RAAKE, A., GARCIA, M. N., AND LIST, P. No-reference video quality assessment for SD and HD H.264/AVC sequences based on continuous estimates of packet loss visibility. In *IEEE QoMEX* (2011).
- [37] AVCIBAS, I., SANKUR, B., AND SAYOOD, K. Statistical evaluation of image quality measures. *Journal of Electronic Imaging* 11, 2 (Apr. 2002), 206–223.
- [38] AVERY, D. LTE quality of experience starts at the base. <http://blogs.ixiacom.com/ixia-blog/validate-lte-base-station-performance/>, 2012.
- [39] AWEYA, J. On the design of IP routers part 1: Router architectures. *Journal of Systems Architecture* 46, 6 (2000), 483–511.
- [40] BALAMASH, A., AND KRUNZ, M. An overview of web caching replacement algorithms. *IEEE Communications Surveys Tutorials* 6, 2 (2004), 44–56.
- [41] BARAKAT, N., AND DARCIE, T. E. Delay characterization of cable access networks. *IEEE Communications Letters* 11, 4 (2007), 357–359.
- [42] BARFORD, P. *Modeling, Measurement and Performance of World Wide Web Transactions*. PhD thesis, Boston University, 2000.
- [43] BARISH, G., AND OBRACZKE, K. World wide web caching: trends and techniques. *IEEE Communications Magazine* 38, 5 (2000), 178–184.

- [44] BEBEN, A. EQ-BGP: an efficient inter-domain QoS routing protocol. In *International Conference on Advanced Information Networking and Applications* (2006).
- [45] BEHESHTI, N., GANJALI, Y., GHOBADI, M., MCKEOWN, N., AND SALMON, G. Experimental study of router buffer sizing. In *ACM IMC* (2008), pp. 197–210.
- [46] BOLOT, J. End-to-end packet delay and loss behavior in the internet. In *ACM SIGCOMM* (1993).
- [47] BONALD, T. The erlang model with non-poisson call arrivals. *SIGMETRICS Perform. Eval. Rev.* 34, 1 (June 2006).
- [48] BORGNAT, P., DEWAELE, G., FUKUDA, K., ABRY, P., AND CHO, K. Seven years and one day: Sketching the evolution of Internet traffic. In *IEEE INFOCOM* (2009).
- [49] BOUCADAIR, M. QoS-enhanced border gateway protocol. <http://tools.ietf.org/html/draft-boucadair-qos-bgp-spec-01>, 2005. Internet Draft (expired).
- [50] BOULOS, F., PARREIN, B., LE CALLET, P., AND HANDS, DAVID, S. Perceptual effects of packet loss on H.264/AVC encoded videos. In *Workshop on Video Processing and Quality Metrics for Consumer Electronics* (2009).
- [51] BRESLAU, L., CAO, P., FAN, L., PHILLIPS, G., AND SHENKER, S. Web caching and Zipf-like distributions: Evidence and implications. In *IEEE INFOCOM* (1999).
- [52] BRUTLAG, J. Speed matters for Google web search. http://services.google.com/fh/files/blogs/google_delayexp.pdf, 2009.
- [53] BUCHINGER, S., AND HLAVACS, H. Subjective quality of mobile MPEG-4 videos with different frame rates. *Journal of Mobile Multimedia* 1, 4 (2006), 327–341.
- [54] CHA, M., KWAK, H., RODRIGUEZ, P., AHN, Y.-Y., AND MOON, S. I tube, you tube, everybody tubes: analyzing the world’s largest user generated content video system. In *ACM IMC* (2007).
- [55] CHENG, K., AND KAMBAYASHI, Y. LRU-SP: a size-adjusted and popularity-aware LRU replacement algorithm for web caching. In *Computer Software and Applications Conference* (2000).
- [56] CHIANG, W.-H., XIAO, W.-C., AND CHOU, C.-F. A performance study of VoIP applications: MSN vs. Skype. In *IEEE ICC MultiComm Workshop* (2006).

-
- [57] CHIKKERUR, S., SUNDARAM, V., REISSLEIN, M., AND KARAM, L. Objective video quality assessment methods: A classification, review, and performance comparison. *IEEE Transactions on Broadcasting* 57, 2 (2011), 165–182.
- [58] CHIOU, J.-S. The antecedents of consumers’ loyalty toward internet service providers. *Information and Management* 41, 6 (July 2004), 685–695.
- [59] CHIRICHELLA, C., AND ROSSI, D. To the moon and back: are Internet bufferbloat delays really that large. In *IEEE INFOCOM Workshop on Traffic Monitoring and Analysis* (2013).
- [60] CLAYPOOL, M., FINKEL, D., GRANT, A., AND SOLANO, M. Thin to win? network performance analysis of the OnLive thin client game system. In *IEEE NetGames* (2012).
- [61] COHEN, R., AND SHOCHOT, A. The ”global-ISP” paradigm. *Computer Networks* 51, 8 (June 2007), 1908–1921.
- [62] COLE, R., AND ROSENBLUTH, J. Voice over IP performance monitoring. *ACM CCR* 4, 3 (Apr. 2001).
- [63] COULL, S., MONROSE, F., REITER, M., AND BAILEY, M. The challenges of effectively anonymizing network data. In *Conference For Homeland Security* (2009).
- [64] COVERDALE, P., MÖLLER, S., RAAKE, A., AND TAKAHASHI, A. Multimedia quality assessment standards in itu-t sg12. *IEEE Signal Processing Magazine* 28, 6 (2011), 91–97.
- [65] CROVELLA, M., AND KRISHNAMURTHY, B. *Internet Measurement*, 1 ed. Wiley, 2006.
- [66] CROVELLA, M. E., AND BESTAVROS, A. Self-similarity in world wide web traffic: evidence and possible causes. In *ACM SIGMETRICS* (1996).
- [67] CUI, H., AND BIRSACK, E. Trouble shooting interactive web sessions in a home environment. In *ACM SIGCOMM Workshop on Home Networks* (2011).
- [68] DAN, A., AND TOWSLEY, D. An approximate analysis of the LRU and FIFO buffer replacement schemes. *SIGMETRICS Perf. Eval. Review* 18, 1 (1990), 143–152.
- [69] DAVISON, B. Brian d. davison’s web caching bibliography. <http://www.web-caching.com/biblio.html>, 2006.
- [70] DISCHINGER, M., HAEBERLEN, A., GUMMADI, K. P., AND SAROIU, S. Characterizing residential broadband networks. In *ACM IMC* (2007).
- [71] EGGER, S., HOSSFELD, T., SCHATZ, R., AND FIEDLER, M. Tutorial: Waiting Times in Quality of Experience for Web based Services. In *IEEE QoMEX* (2012).

- [72] EGGER, S., SCHATZ, R., AND SCHERER, S. It Takes Two to Tango - Assessing The Impact of Delay on Conversational Interactivity on Perceived Speech Quality. In *Interspeech* (2010).
- [73] EGGER, S., SCHATZ, R., SCHOENENBERG, K., RAAKE, A., AND KUBIN, G. Same but different? - using speech signal features for comparing conversational voip quality studies. In *IEEE ICC* (2012).
- [74] ELLIOTT, E. O. Estimates of Error Rates for Codes on Burst-Noise Channels. *Bell System Technical Journal* 42 (Sept. 1963), 1977–1997.
- [75] EMARKSOFTS. Fast email harvester 1.2. <http://fast-email-harvester.smartcode.com/info.html>, 2009.
- [76] ENACHESCU, M., GANJALI, Y., GOEL, A., MCKEOWN, N., AND ROUGHGARDEN, T. Routers with very small buffers. In *IEEE INFOCOM* (2006).
- [77] ENGELKE, U., AND ZEPERNICK, H.-J. Perceptual-based quality metrics for image and video services: A survey. In *EuroNGI* (2007).
- [78] ERIKSSON, B., BARFORD, P., MAGGS, B., AND NOWAK, R. Posit: a lightweight approach for IP geolocation. *SIGMETRICS Perf. Eval. Review* 40, 2 (Oct. 2012).
- [79] ERMAN, J., GERBER, A., HAJIAGHAYI, M., PEI, D., SEN, S., AND SPATSCHECK, O. To cache or not to cache: The 3G case. *IEEE Internet Computing* 15, 2 (2011), 27–34.
- [80] ERMAN, J., GERBER, A., HAJIAGHAYI, M. T., PEI, D., AND SPATSCHECK, O. Network-aware forward caching. In *International World Wide Web Conference* (2009).
- [81] ERMAN, J., GERBER, A., RAMADRISHNAN, K. K., SEN, S., AND SPATSCHECK, O. Over the top video: the gorilla in cellular networks. In *ACM IMC* (2011).
- [82] ETSI. ETSI EN 302 304 – digital video broadcasting (DVB); transmission system for handheld terminals (DVB-H), Nov. 2004.
- [83] EUROIX. List of 306 known IXPS around the globe. <https://www.euro-ix.net/resources-list-of-ixps>, 2012.
- [84] FAIRHURST, G., AND BRISCOE, B. Advice on network buffering. <http://www.ietf.org/proceedings/86/slides/slides-86-iccr-4.pdf>, Mar. 2013.
- [85] FAIRHURST, G., AND BRISCOE, B. Advice on network buffering draft-fairhurst-tsvwg-buffers-00. <http://tools.ietf.org/html/draft-fairhurst-tsvwg-buffers-00>, Mar. 2013.

-
- [86] FAN, J., XU, J., AMMAR, M. H., AND MOON, S. B. Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme. *Computer Networks* 46, 2 (Oct. 2004), 253–272.
- [87] FARIAS, M. C. Q., AND MITRA, S. K. No-reference video quality metric based on artifact measurements. In *IEEE Transactions on Image Processing* (2005).
- [88] FEAMSTER, N., AND BALAKRISHNAN, H. Packet loss recovery for streaming video. In *International Packet Video Workshop* (2002).
- [89] FIEDLER, H. The impact of delay on online gaming: A case study with minecraft. Master’s thesis, Technische Universität Berlin, Dec. 2013.
- [90] FIEDLER, M., HOSSFELD, T., AND TRAN-GIA, P. A generic quantitative relationship between Quality of Experience and Quality of Service. *Netw. Mag. of Global Internetwkg.* 24, 2 (Mar. 2010), 36–41.
- [91] FLOYD, S., AND JACOBSON, V. Random early detection gateways for congestion avoidance. *IEEE/ACM ToN* 1, 4 (Aug. 1993), 397–413.
- [92] FOFACK, N., NAIN, P., NEGLIA, G., AND TOWSLEY, D. Analysis of TTL-based cache networks. In *Conference on Performance Evaluation Methodologies and Tools* (2012).
- [93] FROSSARD, P. FEC performances in multimedia streaming. *IEEE Communications Letters* 5, 3 (2001), 122–124.
- [94] GARCIA, M., AND RAAKE, A. Parametric packet-layer video quality model for IPTV. In *ISSPA* (2010).
- [95] GELENBE, E. A unified approach to the evaluation of a class of replacement algorithms. *IEEE Trans. Comput.* 22, 6 (June 1973), 611–618.
- [96] GERBER, A., AND DOVERSPIKE, R. Traffic types and growth in backbone networks. In *Optical Fiber Communication Conference* (2011).
- [97] GETTYS, J. IW10 considered harmful. IETF Internet-Draft, 2011.
- [98] GETTYS, J., AND NICHOLS, K. Bufferbloat: Dark buffers in the internet. *ACM Queue* 9 (Nov. 2011), 40:40–40:54.
- [99] GHOBADI, M., CHENG, Y., JAIN, A., AND MATHIS, M. Trickle: Rate limiting youtube video streaming. In *USENIX ATC* (2012).
- [100] GILBERT, E. N. Capacity of a Burst-Noise Channel. *Bell System Technical Journal* 39 (Sept. 1960), 1253–1265.
- [101] GILL, P., ARLITT, M. F., LI, Z., AND MAHANTI, A. Youtube traffic characterization: a view from the edge. In *ACM IMC* (2007).

- [102] GIROD, B., STUHLMÜLLER, K., LINK, M., AND HORN, U. Packet loss resilient internet video streaming. In *SPIE Visual Communications and Image Processing* (1999).
- [103] GOLLE, P. Revisiting the uniqueness of simple demographics in the US population. In *ACM WPES* (2006).
- [104] GOMES, L. H., CAZITA, C., ALMEIDA, J. M., ALMEIDA, V., AND MEIRA, JR., W. Characterizing a spam traffic. In *ACM IMC* (2004), pp. 356–369.
- [105] GONG, Y., ROSSI, D., TESTA, C., VALENTI, S., AND TAHT, D. Fighting the bufferbloat: on the coexistence of AQM and low priority congestion control. In *IEEE INFOCOM Workshop on Traffic Monitoring and Analysis* (2013).
- [106] GREENGRASS, J., EVANS, J., AND BEGEN, A. C. Not all packets are equal, part 2: The impact of network packet loss on video quality. *IEEE Internet Computing* 13, 2 (2009), 74–82.
- [107] GROUP, M. M. World internet users statistics usage and world population stats. <http://www.internetworldstats.com/stats.htm>, 2012.
- [108] GU, G., ZHANG, J., AND LEE, W. BotSniffer: Detecting botnet command and control channels in network traffic. In *Network and Distributed System Security Symposium* (2008).
- [109] GU, Y., TOWSLEY, D. F., HOLLOT, C. V., AND ZHANG, H. Congestion control for small buffer high speed networks. In *IEEE INFOCOM* (2007).
- [110] GUSE, D., AND MÖLLER, S. Long-term impact of varying multimedia service performance on quality ratings in a multiservice scenario. In *DAGA* (2012).
- [111] GUSE, D., AND MÖLLER, S. Macro-temporal development of QoE: Impact of varying performance on QoE over multiple interactions. In *DAGA* (2013).
- [112] HAIQING JIANG, YAOGONG WANG, K. L., AND RHEE, I. Tackling bufferbloat in 3G/4G networks. In *ACM IMC* (2012).
- [113] HAMMER, F., REICHL, P., AND RAAKE, A. The Well-Tempered Conversation: Interactivity, Delay and Perceptual VoIP Quality. In *IEEE ICC* (2005).
- [114] HAO, S., SYED, N. A., FEAMSTER, N., GRAY, A. G., AND KRASSER, S. Detecting spammers with SNARE: spatio-temporal network-level automatic reputation engine. In *USENIX Security Symposium* (2009).
- [115] HASSLINGER, G., AND HOHLFELD, O. Efficiency of caches for content distribution on the internet. In *International Teletraffic Congress* (2010).
- [116] HEMMINGER, S. Network emulation with NetEm. In *Australia’s National Linux Conference* (2005).

-
- [117] HOHLFELD, O. Statistical error model to impair an H.264 decoder. Master's thesis, Technische Universität Darmstadt, Mar. 2008. KOM-D-0310.
- [118] HOHLFELD, O., BALARAJAH, B., BENNER, S., RAAKE, A., AND CIUCU, F. On revealing the arq mechanism of mstv. In *Proceedings of IEEE International Conference on Communications (ICC '11)* (June 2011), IEEE.
- [119] HOHLFELD, O., AND CIUCU, F. Viewing impaired video transmissions from a modeling perspective. *Performance Evaluation Review* 37, 2 (Sept. 2009), 33–35.
- [120] HOHLFELD, O., GRAF, T., AND CIUCU, F. Longtime Behavior of Harvesting Spam Bots. In *Internet Measurement Conference (2012)*.
- [121] HOHLFELD, O., PUJOL, E., CIUCU, F., FELDMANN, A., AND BARFORD, P. Bufferbloat: How relevant? a QoE perspective on buffer sizing. Tech. Rep. 2012-11, Department of Computer Science, Technische Universität Berlin, 2012.
- [122] HOLZ, T., GORECKI, C., RIECK, K., AND FREILING, F. C. Measuring and detecting fast-flux service networks. In *Network and Distributed System Security Symposium (2008)*.
- [123] HONG, D., VLEESCHAUWER, D., AND BACCELLI, F. A chunk-based caching algorithm for streaming video. In *Workshop on Network Control and Optimization (2010)*.
- [124] HOSSFELD, T., BIEDERMANN, S., SCHATZ, R., PLATZER, A., EGGER, S., AND FIEDLER, M. The memory effect and its implications on web QoE modeling. In *International Teletraffic Congress (2011)*, pp. 103–110.
- [125] HOSSFELD, T., EGGER, S., SCHATZ, R., FIEDLER, M., MASUCH, K., AND LORENTZEN, C. Initial delay vs. interruptions: Between the devil and the deep blue sea. In *IEEE QoMEX (2012)*.
- [126] HOSSFELD, T., HOCK, D., TRAN-GIA, P., TUTSCHKU, K., AND FIEDLER, M. Testing the IQX hypothesis for exponential interdependency between QoS and QoE of voice codecs iLBC and G.711. In *ITC Specialist Seminar on Quality of Experience (2008)*.
- [127] HOSSFELD, T., SCHATZ, R., AND EGGER, S. SOS: The MOS is not enough! In *IEEE QoMEX (2011)*.
- [128] HOSSFELD, T., STROHMEIER, D., RAAKE, A., AND SCHATZ, R. Pippi long-stocking calculus for temporal stimuli pattern on YouTube QoE: $1 + 1 = 3$ and $1 \times 4 \neq 4 \times 1$. In *Workshop on Mobile Video (2013)*.
- [129] HOSSFELD, T., TRAN-GIA, P., AND FIEDLER, M. Quantification of Quality of Experience for edge-based applications. In *International Teletraffic Congress (2007)*.

- [130] HU, X., KNYSZ, M., AND SHIN, K. G. Measurement and analysis of global IP-usage patterns of fast-flux botnets. In *IEEE INFOCOM* (2011).
- [131] HUITIKA, T., AND DRIESSEN, P. Datagram loss model for noninteractive real time streaming video. In *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing* (2003).
- [132] HUSTON, G. The 32-bit AS number report. <http://www.potaroo.net/tools/asn32/>, 2012.
- [133] HUSTON, G. BGP statistics from route-views data. <http://bgp.potaroo.net/bgprpts/rva-index.html>, 2012.
- [134] HUYNH-THU, Q., AND GHANBARI, M. Impact of jitter and jerkiness on perceived video quality. In *International Workshop on Video Processing and Quality Metrics for Consumer Electronics* (2006).
- [135] HUYNH-THU, Q., AND GHANBARI, M. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters* 44, 13 (2008).
- [136] JACOBSON, V. Modified TCP congestion control algorithm. End2end-interest mailing list, Apr. 1990.
- [137] JIANG, W., AND SCHULZRINNE, H. Modeling of packet loss and delay and their effect on real-time multimedia service quality. In *ACM NOSSDAV* (2000).
- [138] JIANG, W., AND SCHULZRINNE, H. Comparison and optimization of packet loss repair methods on voip perceived quality under bursty loss. In *ACM NOSSDAV* (2002).
- [139] JOHN, J. P., MOSHCHUK, A., GRIBBLE, S. D., AND KRISHNAMURTHY, A. Studying spamming botnets using botlab. In *USENIX NSDI* (2009).
- [140] JURGELIONIS, A., LAULAJAINEN, J., HIRVONEN, M., AND WANG, A. An empirical study of NetEm network emulation functionalities. In *International Conference on Computer Communications and Networks* (2011).
- [141] KANICH, C., KREIBICH, C., LEVCHENKO, K., ENRIGHT, B., VOELKER, G. M., PAXSON, V., AND SAVAGE, S. Spamalytics: An empirical analysis of spam marketing conversion. In *ACM Conference on Computer and Communications Security* (2008).
- [142] KANUMURI, S., COSMAN, P., AND REIBMAN, A. R. A generalized linear model for MPEG-2 packet-loss visibility. In *International Packet Video Workshop* (2004).
- [143] KANUMURI, S., COSMAN, P. C., REIBMAN, A. R., AND VAISHAMPAYAN, V. A. Modeling packet-loss visibility in MPEG-2 video. *IEEE Transactions on Multimedia* 8, 2 (April 2006), 341–355.

-
- [144] KANUMURI, S., SUBRAMANIAN, S. G., COSMAN, P. C., AND REIBMAN, A. R. Predicting H.264 packet loss visibility using a generalized linear model. In *IEEE Transactions on Image Processing* (2006).
- [145] KARN, P., BORMANN, C., FAIRHURST, G., GROSSMAN, D., LUDWIG, R., MAHDAVI, J., MONTENEGRO, G., TOUCH, J., AND WOOD, L. Advice for Internet subnetwork designers. RFC 3819, July 2004.
- [146] KARN, P., AND PARTRIDGE, C. Improving round-trip time estimates in reliable transport protocols. *ACM CCR* 17, 5 (Aug. 1987).
- [147] KHAN, A., SUN, L., AND IFEACHOR, E. Impact of video content on video quality for video over wireless networks. In *International Conference on Autonomous and Autonomous Systems* (2009).
- [148] KIM, J., CHUNG, K., AND CHOI, K. Spam filtering with dynamically updated URL statistics. *IEEE Security and Privacy* 5, 4 (July 2007), 33–39.
- [149] KING, W. F. Analysis of paging algorithms. In *IFIP Congress* (1971).
- [150] KITAWAKI, N., AND ITOH, K. Pure delay effects on speech quality in telecommunications. *IEEE Journal on Selected Areas in Communications* 9, 4 (1991), 586–593.
- [151] KLAUE, J., RATHKE, B., AND WOLISZ, A. Evalvid - a framework for video transmission and quality evaluation. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation* (2003).
- [152] KLEIN, M. Akamai technologies. <http://www.cs.odu.edu/~mukka/cs775s07/Presentations/mklein.pdf>, 2007.
- [153] KNOCHE, H., AND SASSE, M. A. The big picture on small screens delivering acceptable video quality in mobile TV. *ACM Trans. Multimedia Comput. Commun. Appl.* 5, 3 (2009).
- [154] KNYSZ, M., HU, X., AND SHIN, K. G. Good guys vs. bot guise: Mimicry attacks against fast-flux detection systems. In *IEEE INFOCOM* (2011).
- [155] KORDELAS, A., DAGIUKLAS, T., AND POLITIS, I. On the performance of H.264/AVC over lossy IP-based networks. In *European Signal Processing Conference* (2012).
- [156] KORHONEN, J., AND YOU, J. Peak Signal-to-Noise Ratio revised: Is simple beautiful? In *IEEE QoMEX* (2012).
- [157] KREIBICH, C., KANICH, C., LEVCHENKO, K., ENRIGHT, B., VOELKER, G. M., PAXSON, V., AND SAVAGE, S. Spamcraft: an inside look at spam campaign orchestration. In *USENIX LEET* (2009).
- [158] KREIBICH, C., WEAVER, N., NECHAEV, B., AND PAXSON, V. Netalyzr: Illuminating the edge network. In *ACM IMC* (2010).

- [159] KRENC, T. Measurement and characterization of content distribution in BitTorrent. Master's thesis, Technische Universität Berlin, 2012.
- [160] KRISHNAMOORTHY, V., BERGSTRÖM, P., CARLSSON, N., EAGER, D., MAHANTI, A., AND SHAHMEHRI, N. Empowering the creative user: personalized HTTP-based adaptive streaming of multi-path nonlinear video. In *ACM SIGCOMM Workshop on Future Human-Centric Multimedia Networking* (2013).
- [161] KRISHNAN, S. S., AND SITARAMAN, R. K. Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs. In *ACM IMC* (2012).
- [162] KUROSE, J., AND ROSS, K. *Computer Networking: A Top-Down Approach*, 4 ed. Addison Wesley, 2008.
- [163] LABOVITZ, C., IEKEL-JOHNSON, S., MCPHERSON, D., OBERHEIDE, J., AND JAHANIAN, F. Internet inter-domain traffic. *ACM CCR* 41, 4 (Aug. 2010).
- [164] LAKSHMIKANTHA, A., BECK, C., AND SRIKANT, R. Impact of file arrivals and departures on buffer sizing in core routers. *IEEE/ACM ToN* 19, 2 (Apr. 2011), 347–358.
- [165] LEE, J.-S., SIMONE, F. D., AND EBRAHIMI, T. Subjective quality evaluation via paired comparison: Application to scalable video coding. *IEEE Transactions on Multimedia* 13, 5 (2011), 882–893.
- [166] LEE, J.-S., SIMONE, F. D., EBRAHIMI, T., RAMZAN, N., AND IZQUIERDO, E. Quality assessment of multidimensional video scalability. *IEEE Communications Magazine* 50, 4 (2012), 38–46.
- [167] LELAND, W. E., TAQQU, M. S., WILLINGER, W., AND WILSON, D. V. On the self-similar nature of Ethernet traffic. *ACM CCR* 23, 4 (1993), 183–193.
- [168] LI, M., CHEN, Z., AND TAN, Y.-P. QoE-aware resource allocation for scalable video transmission over multiuser mimo-ofdm systems. In *IEEE VCIP* (2012).
- [169] LI, M., CHEN, Z., AND TAN, Y.-P. On Quality of Experience of scalable video adaptation. *Journal of Visual Communication and Image Representation* 24, 5 (2013), 509 – 521.
- [170] LINDEN, G. Make data useful. <http://de.scribd.com/doc/4970486/Make-Data-Useful-by-Greg-Linden-Amazoncom>, 2006.
- [171] LIU, X., DOBRIAN, F., MILNER, H., JIANG, J., SEKAR, V., STOICA, I., AND ZHANG, H. A case for a coordinated internet video control plane. In *ACM SIGCOMM* (2012).

-
- [172] MAIER, G., FELDMANN, A., PAXSON, V., AND ALLMAN, M. On dominant characteristics of residential broadband internet traffic. In *ACM IMC* (2009), pp. 90–102.
- [173] MARTIN, J., AND NILSSON, A. A. On service level agreements for IP networks. In *IEEE INFOCOM* (2002).
- [174] MARTIN, J., WESTALL, J., SHAW, T., WHITE, G., WOUNDY, R., FINKELSTEIN, J., AND HART, G. Cable modem buffer management in docsis networks. In *IEEE Conference on Sarnoff* (2010).
- [175] MAYER, M. What Google knows. In *Web 2.0 Summit* (2006).
- [176] MAYER, M. In search of... a better, faster, stronger web. In *Velocity* (2009).
- [177] MCCARTHY, J. D., SASSE, M. A., AND MIRAS, D. Sharp or smooth?: comparing the effects of quantization vs. frame rate for streamed video. In *SIGCHI conference* (2004).
- [178] MCDUGALL, J., AND MILLER, S. Sensitivity of wireless network simulations to a two-state Markov model channel approximation. In *IEEE GLOBECOM* (2003).
- [179] MOK, R. K. P., CHAN, E. W. W., AND CHANG, R. K. C. Measuring the Quality of Experience of HTTP video streaming. In *IEEE International Symposium on Integrated Network Management* (2011), pp. 485–492.
- [180] MOK, R. K. P., LUO, X., CHAN, E. W. W., AND CHANG, R. K. C. QDASH: a QoE-aware DASH system. In *ACM Multimedia Systems Conference* (2012).
- [181] MÖLLER, S., BANG, C., TAMME, T., VAALGAMAA, M., AND WEISS, B. From single-call to multi-call quality: A study on long-quality integration in audio-visual speech communication. In *INTERSPEECH* (2011).
- [182] MÖLLER, S., CHAN, W.-Y., CÔTÉ, N., FALK, T. H., RAAKE, A., AND WÄLTERMANN, M. Speech quality estimation: Models and trends. *IEEE Signal Processing Magazine* 28, 6 (2011).
- [183] MÖLLER, S., CHAN, W.-Y., CÔTÉ, N., FALK, T. H., RAAKE, A., AND WÄLTERMANN, M. Speech quality estimation: Models and trends. *IEEE Signal Process. Mag.* 28, 6 (2011), 18–28.
- [184] MOON, S. B., KUROSE, J., AND TOWSLEY, D. Packet audio playout delay adjustments: performance bounds and algorithms. *Multimedia Systems* 6 (1998), 17–28.
- [185] MÜLLER, C., AND TIMMERER, C. A VLC media player plugin enabling dynamic adaptive streaming over HTTP. In *ACM international conference on Multimedia* (2011).

- [186] NEMETHOVA, O., RIES, M., ZAVODSKY, M., AND RUPP, M. PSNR-based estimation of subjective time-variant video quality for mobiles. In *International Conference on Measurement of Speech, Audio and Video Quality in Networks* (2006).
- [187] NICHOLS, K., AND JACOBSON, V. Controlling queue delay. *Commun. ACM* 55, 7 (July 2012), 42–50.
- [188] NICHOLS, K., AND JACOBSON, V. Controlling queue delay. *ACM Queue* 10, 5 (May 2012).
- [189] NORTHWORKS SOLUTIONS LTD. Ecrawl v2.63. <http://www.northworks.biz/software.html>, 2012.
- [190] OECD. Spam issues in developing countries. <http://www.oecd.org/internet/ieconomy/34935342.pdf>, 2005.
- [191] ONG, E., YANG, X., LIN, W., LU, Z., AND YAO, S. Video quality metric for low bitrate compressed videos. In *IEEE Transactions on Image Processing* (2004).
- [192] OSTERMANN, J., BORMANS, J., LIST, P., MARPE, D., NARROSCHE, M., PEREIRA, F., STOCKHAMMER, T., AND WEDI, T. Video coding with H.264/AVC: tools, performance, and complexity. *IEEE Circuits and Systems Magazine* 4, 1 (2004), 7–28.
- [193] PANG, R., ALLMAN, M., PAXSON, V., AND LEE, J. The devil and packet trace anonymization. *ACM CCR* 36, 1 (Jan. 2006), 29–38.
- [194] PAPER, P. W. Why QoE is important to service providers. http://www.peerapp.com/App_FCK/file/PeerApp_QoE_White_Paper.pdf, 2008.
- [195] PARANDEHGHEIBI, A., MEDARD, M., OZDAGLAR, A., AND SHAKKOTTAI, S. Access-network association policies for media streaming in heterogeneous environments. *ArXiv e-prints* (Apr. 2010). 1004.3523.
- [196] PASTRANA-VIDAL, R. R., AND GICQUEL, J.-C. Automatic quality assessment of video fluidity impairments using a no-reference metric. In *Int. Workshop on Video Processing and Quality Metrics for Consumer Electronics* (2006).
- [197] PATHAK, A., QIAN, F., HU, Y. C., MAO, Z. M., AND RANJAN, S. Botnet spam campaigns can be long lasting: evidence, implications, and analysis. In *ACM SIGMETRICS* (2009).
- [198] PATRICK SEELING, H. F., AND REISSLEIN, M. *Video Traces for Network Performance Evaluation Video Traces for Network Performance Evaluation*. Springer, 2007.

-
- [199] PAXSON, V. Bro: a system for detecting network intruders in real-time. *Computer Networks* 31, 23-24 (1999), 2435–2463.
- [200] PAXSON, V., AND FLOYD, S. Wide area traffic: the failure of poisson modeling. *IEEE/ACM ToN* 3, 3 (June 1995), 226–244.
- [201] PINSON, M., AND WOLF, S. A new standardized method for objectively measuring video quality. *IEEE Transactions on Broadcasting* 50, 3 (2004), 312–322.
- [202] PLISSONNEAU, L., COSTEUX, J.-L., AND BROWN, P. Analysis of Peer-to-Peer traffic on ADSL. In *PAM* (2005).
- [203] PODLIPNIG, S., AND BÖSZÖRMENYI, L. A survey of web cache replacement strategies. *ACM Comput. Surv.* 35, 4 (Dec. 2003), 374–398.
- [204] POESE, I., FRANK, B., AGER, B., SMARAGDAKIS, G., AND FELDMANN, A. Improving content delivery using provider-aided distance information. In *ACM IMC* (2010).
- [205] POESE, I., FRANK, B., AGER, B., SMARAGDAKIS, G., UHLIG, S., AND FELDMANN, A. Improving content delivery with PaDIS. *IEEE Internet Computing* (2012).
- [206] POESE, I., UHLIG, S., KAAFAR, M. A., DONNET, B., AND GUEYE, B. Ip geolocation databases: unreliable? *ACM CCR* 41, 2 (Apr. 2011).
- [207] POIKONEN, J. Half-normal run length packet channel models applied in DVB-H simulations. In *Symposium on Wireless Communication Systems* (2006).
- [208] PRASAD, R. S., DOVROLIS, C., AND THOTTAN, M. Router buffer sizing for tcp traffic and the role of the output/input capacity ratio. *IEEE/ACM ToN* 17, 5 (Oct. 2009), 1645–1658.
- [209] PRINCE, M. B., DAHL, B. M., HOLLOWAY, L., KELLER, A. M., AND LANGHEINRICH, E. Understanding how spammers steal your e-mail address: An analysis of the first six months of data from project honey pot. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference* (2005).
- [210] PRINS, M. J., BRUNNER, M., KARAGIANNIS, G., LUNDQVIST, H., AND NUNZI, G. Fast RTP retransmission for IPTV – implementation and evaluation. In *IEEE GLOBECOM* (2008).
- [211] QIAN, F., QUAH, K. S., HUANG, J., ERMAN, J., GERBER, A., MAO, Z., SEN, S., AND SPATSCHECK, O. Web caching on smartphones: ideal vs. reality. In *Conference on Mobile systems, applications, and services* (2012).
- [212] RAAKE, A. Predicting speech quality under random packet loss: Individual impairment and additivity with other network impairments. *Acta Acustica* 90 (2004), 1061–1083.

- [213] RAAKE, A. *Speech Quality of VoIP: Assessment and Prediction*. Wiley, 2006.
- [214] RAAKE, A. et al. Definition of Quality of Experience. In *Qualinet White paper on Definitions of Quality of Experience*, S. M. Patrick Le Callet and A. Perkis, Eds. 2012.
- [215] RAAKE, A., GARCIA, M., MOELLER, S., BERGER, J. AND KLING, F., LIST, P., JOHANN, J., AND HEIDEMANN, C. T-V-model: Parameter-based prediction of IPTV quality. In *ISSPA* (2008).
- [216] RABINER, L. R. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE* 77, 2 (1989), 257–286.
- [217] RAMACHANDRAN, A., AND FEAMSTER, N. Understanding the network-level behavior of spammers. In *ACM SIGCOMM* (2006).
- [218] RAMJEE, R., KUROSE, J. F., TOWSLEY, D. F., AND SCHULZRINNE, H. Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *IEEE INFOCOM* (1994).
- [219] REIBMAN, A. R., AND POOLE, D. Characterizing packet-loss impairments in compressed video. In *IEEE Transactions on Image Processing* (2007).
- [220] REICHL, P., EGGER, S., SCHATZ, R., AND D’ALCONZO, A. The logarithmic nature of QoE and the role of the Weber-Fechner law in QoE assessment. In *IEEE ICC* (2010).
- [221] ROSENBERG, J., AND SCHULZRINNE, H. Integrating packet FEC into adaptive voice playout buffer algorithms on the Internet. In *IEEE INFOCOM* (2000).
- [222] SAT, B., AND WAH, B. W. Analyzing voice quality in popular VoIP applications. *IEEE MultiMedia* 16, 1 (2009), 46–59.
- [223] SCHRYEN, G. An e-mail honeypot addressing spammers’ behavior in collecting and applying addresses. In *IEEE Information Assurance Workshop* (2005).
- [224] SCHULZE, H., AND MOCHALSKI, K. Internet study 2007. <http://www.ipoque.com/sites/default/files/mediafiles/documents/internet-study-2007.pdf>, 2007.
- [225] SCHULZE, H., AND MOCHALSKI, K. Internet study 2008/2009. <http://www.ipoque.com/sites/default/files/mediafiles/documents/internet-study-2008-2009.pdf>, 2009.
- [226] SCHURMAN, E., AND BRUTLAG, J. Performance related changes and their user impact. In *Velocity* (2009). <http://www.youtube.com/watch?v=bQSE51-gr2s>.
- [227] SESHADRINATHAN, K., SOUNDARARAJAN, R., BOVIK, A. C., AND CORMACK, L. K. Study of subjective and objective quality assessment of video. *IEEE Transactions on Image Processing* 19, 6 (June 2010), 1427–1441.

-
- [228] SHUE, C. A., GUPTA, M., LUBIA, J. J., KONG, C. H., AND YUKSEL, A. Spamology: A study of spam origins. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference* (2009).
- [229] SIEBER, C., HOSSFELD, T., ZINNER, T., TRAN-GIA, P., AND TIMMERER, C. Implementation and user-centric comparison of novel adaption logic for DASH with SVC. In *IFIP / IEEE Workshop on Quality of Experience Centric Management* (2013).
- [230] SILBERSCHATZ, A., AND GALVIN, P. B. *Operating Systems Concepts*. Addison Wesley, 1994.
- [231] SIVAPRAKASAM, S., AND SHANMUGAN, K. An Equivalent Markov Model for Burst Errors in Digital Channels. *IEEE Transactions on Communications* 43, 234 (Feb/Mar/Apr 1995), 1347–1355.
- [232] SKOWRONEK, J., HERLINGHAUS, J., AND RAAKE, A. Quality assessment of asymmetric multiparty telephone conferences: A systematic method from technical degradations to perceived impairments. In *Interspeech* (2013).
- [233] SOMMERS, J., BARFORD, P., GREENBERG, A., AND WILLINGER, W. An SLA perspective on the router buffer sizing problem. *SIGMETRICS Perf. Eval. Review* 35 (March 2008), 40–51.
- [234] SOMMERS, J., KIM, H., AND BARFORD, P. Harpoon: A flow-level traffic generator for router and network tests. *SIGMETRICS Perf. Eval. Review* 32, 1 (June 2004), 392–392.
- [235] SPIRA, J. B. Spam e-mail and its impact on it spending and productivity. [http://www.basex.com/poty2003.nsf/e67dc0f5617d6e9c85256a99005ea0e7/f8761f74ba37069385256e040019f314/\\$FILE/BasexReport.Spam.pdf](http://www.basex.com/poty2003.nsf/e67dc0f5617d6e9c85256a99005ea0e7/f8761f74ba37069385256e040019f314/$FILE/BasexReport.Spam.pdf), 2003.
- [236] STAROBINSKI, D., AND TSE, D. N. C. Probabilistic methods for web caching. *SIGMETRICS Perf. Eval. Review* 46, 2-3 (2001), 125–137.
- [237] STEFANOV, S. Yslow 2.0. In *China Software Developers Network* (2008). <http://www.slideshare.net/stoyan/yslow-20-presentation>.
- [238] STROHMEIER, D., JUMISKO-PYYKKÖ, S., AND RAAKE, A. Toward task-dependent evaluation of web-QoE: Free exploration vs. ”who ate what?”. In *IEEE GLOBECOM* (2012).
- [239] SUN, L., AND IFEACHOR, E. New models for perceived voice quality prediction and their application in playout buffer optimization for VoIP networks. In *IEEE ICC* (2004).

- [240] SUNDARESAN, S., DE DONATO, W., FEAMSTER, N., TEIXEIRA, R., CRAWFORD, S., AND PESCAPE, A. Measuring home broadband performance. *Commun. ACM* 55, 11 (Nov. 2012), 100–109.
- [241] SYSTEMS, C. Forecast and methodology, 2011-2016. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper/c11-481360_ns827_Networking_Solutions_White_Paper.html, 2012.
- [242] TANG, C., AND MCKINLEY, P. K. Modeling multicast packet losses in wireless LANs. In *ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (2003).
- [243] THE RADICATI GROUP, INC. Email statistics report, 2013 - 2017. <http://www.radicati.com/?p=9659>, 2013.
- [244] THU, Q. H., AND GHANBARI, M. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters* 44, 13 (June 2008), 800–801.
- [245] TRACZINSKI, D., HASSLINGER, G., AND LUTHER, W. Polynomial factorization for servers with semi-markovian workload: performance and numerical aspects of a verified solution technique. *Stochastic Models* 21 (2005), 643–668.
- [246] TSYBAKOV, B., AND GEORGANAS, N. D. On self-similar traffic in ATM queues: definitions, overflow probability bound, and cell delay distribution. *IEEE/ACM ToN* 5, 3 (1997), 397–409.
- [247] TURIN, W. *Performance Analysis and Modeling of Digital Transmission Systems (Information Technology: Transmission, Processing and Storage)*. Springer New York, Inc., Secaucus, NJ, USA, 2004.
- [248] UKAI, Y., AND TAKEMURA, T. Spam mails impede economic growth. *The Review of Socionetwork Strategies* 1, 1 (2007), 14–22.
- [249] UKHANOVA, A., KORHONEN, J., AND FORCHHAMMER, S. Objective assessment of the impact of frame rate on video quality. In *ICIP* (2012).
- [250] VILLAMIZAR, C., AND SONG, C. High performance TCP in ANSNET. *ACM CCR* 24, 5 (Oct. 1994), 45–60.
- [251] VISHWANATH, A., SIVARAMAN, V., AND THOTTAN, M. Perspectives on router buffer sizing: recent results and open problems. *ACM CCR* 39, 2 (Mar. 2009), 34–39.
- [252] VISHWANATH, A., SIVARAMAN, V., ZHAO, Z., RUSSELL, C., AND THOTTAN, M. Adapting router buffers for energy efficiency. In *ACM CoNEXT* (2011).
- [253] WANG, J. A survey of web caching schemes for the internet. *ACM CCR* 29, 5 (Oct. 1999), 36–46.

-
- [254] WANG, M., AND GANJALI, Y. The effects of fairness in buffer sizing. In *IFIP-TC6 conference on Ad Hoc and sensor networks, wireless networks, next generation internet* (2007).
- [255] WANG, Z., AND BOVIK, A. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine* 26, 1 (2009), 98–117.
- [256] WANG, Z., BOVIK, A. C., SHEIKH, H. R., AND SIMONCELLI, E. P. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13 (2004), 600–612.
- [257] WANG, Z., AND LI, Q. Video quality assessment using a statistical model of human visual speed perception. *Journal of the Optical Society of America A - Optics, Image Science and Vision* 24, 12 (2007), B61–B69.
- [258] WANG, Z., LU, L., AND BOVIK, A. Video quality assessment based on structural distortion measurement. *Signal Processing: Image Communication* 19, 1 (2004).
- [259] WANG, Z., SIMONCELLI, E., AND BOVIK, A. Multiscale structural similarity for image quality assessment. In *Asilomar Conf on Signals, Systems and Computers* (2003).
- [260] WEISS, B., MÖLLER, S., RAAKE, A., BERGER, J., AND ULLMANN, R. Modeling call quality for time-varying transmission characteristics using simulated conversational structures. *Acta Acustica united with Acustica* 95, 6 (2009), 11401151.
- [261] WILLINGER, W., TAQQU, M. S., SHERMAN, R., AND WILSON, D. V. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. In *ACM SIGCOMM* (1995).
- [262] WINKLER, S. Video quality and beyond. In *European Signal Processing Conference* (2007).
- [263] WOBKER, L. J. Power consumption in high-end routing systems. NANOG54: <http://www.nanog.org/meetings/nanog54/presentations/Wednesday/Wobker.pdf> http://www.nanog.org/meetings/nanog54/presentations/Wednesday/NANOG_54_04_Power_Consumption.wmv, 2012.
- [264] XIAO, L., LUI, K.-S., WANG, J., AND NAHRSTED, K. QoS extension to BGP. In *IEEE International Conference on Network Protocols* (2002).
- [265] XIE, Y., YU, F., ACHAN, K., PANIGRAHY, R., HULTEN, G., AND OSIPKOV, I. Spamming botnets: signatures and characteristics. *ACM CCR* 38, 4 (Oct. 2008), 171–182.

- [266] XU, K. S., KLIGER, M., CHEN, Y., WOOLF, P. J., AND HERO, A. O. Revealing social networks of spammers through spectral clustering. In *IEEE ICC* (2009).
- [267] XU, Q., HUANG, J., WANG, Z., QIAN, F., GERBER, A., AND MAO, M. Cellular data network infrastructure characterization and implication on mobile content placement. In *ACM SIGMETRICS* (2011).
- [268] ZHANG, L., CHOW, D., AND NG, C. H. Cell loss effect on QoS for MPEG video transmission in ATM networks. In *IEEE ICC* (1999).
- [269] ZHANG, L., ZHU, J., AND YAO, T. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing* 3, 4 (Dec. 2004), 243–269.
- [270] ZHANG, T., CUI, Y., ZHAO, Y., FU, L., AND KORKMAZ, T. Scalable bgp qos extension with multiple metrics. In *International conference on Networking and Services* (2006).
- [271] ZHAO, Y., XIE, Y., YU, F., KE, Q., YU, Y., CHEN, Y., AND GILLUM, E. Botgraph: large scale spamming botnet detection. In *USENIX NSDI* (2009).
- [272] ZHUANG, L., DUNAGAN, J., SIMON, D. R., WANG, H. J., AND TYGAR, J. D. Characterizing botnets from email spam records. In *USENIX LEET* (2008).
- [273] ZINK, M. *Scalable Internet Video-on-Demand Systems*. PhD thesis, Darmstadt University of Technology, July 2003.
- [274] ZINNER, T., ABOUD, O., HOHLFELD, O., AND HOSSFELD, T. Impact of frame rate and resolution on objective QoE metrics. In *IEEE Workshop on Quality of Multimedia Experience* (2010).
- [275] ZINNER, T., ABOUD, O., HOHLFELD, O., HOSSFELD, T., AND TRAN-GIA, P. Towards qoe management for scalable video streaming. In *ITC Specialist Seminar on Multimedia Applications - Traffic, Performance and QoE* (2010).
- [276] ZORZI, M. Performance of FEC and ARQ error control in bursty channels under delay constraints. In *IEEE Vehicular Technology Conference* (1998).