Bjørn J. Villa

Enhancing Quality Aspects of Adaptive Video Streaming in Home Networks

Thesis for the degree of Philosophiae Doctor

Trondheim, May 2014

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics
and Electrical Engineering
Department of Electronics and Telecommunications

NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology, Mathematics and
Electrical Engineering
Department of Electronics and Telecommunications

ISBN 978-82-471-4935-5 (printed ver.)
ISBN 978-82-471-4936-2 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2014:13

Printed by NTNU-trykk

Abstract

The growth of broadband Internet access across the world for the last 10 years has made new business models possible. Services which before mainly were provided by network operators and in dedicated networks have now migrated to the open Internet. This has created a global service provider market, using the Internet as platform. The network operators are in many cases left with only providing the broadband access service. For the global service providers, continuous effort is put into the field of finding smart methods for bringing new and advanced services to the market without asking for e.g. QoS features from the involved network operators. This type of service delivery is called Over-The-Top (OTT). The concept of making services able to adapt their network and transport requirements during time of delivery is a strong contribution to success for OTT services.

The focus of the work in this thesis, is methods for improving various aspects - as defined by my research questions - related to QoS and potentially also QoE for dynamic adaptive video streaming over HTTP (DASH) services. The motivation for focusing on video services is based on their high QoS requirements (i.e. bandwidth) and also popularity in terms of usage.

In the work presented in this thesis, I have studied the behaviour and performance of DASH services by means of simulations, measurements and experiments. Based on insight obtained through this, I established a hypothesis on how QoS and QoE aspects could be improved for users present in the same home network environment. My hypothesis was that making more accurate information available about both services and network conditions in near real-time could facilitate improved control methods. The effectiveness of my suggested methods have in most cases been analysed by means of implementation in an experimental lab scenario, and supported by simulations and analytical approaches when appropriate.

The findings presented in the included papers are all closely related and map into the research model used. This model is composed of Knowledge Plane, Monitor Plane and Action plane components located in both service endpoints and involved network components. As my main focus has been on OTT service delivery, the main contributions of my work apply to service endpoints, i.e. components in the home network and on the server side. The server side would in many cases be represented by a Content Delivery Network (CDN) node.

The main research questions identified are:

RQ1: In a home network how to (autonomously) control the performance of DASH based services.

RQ2: How to provide fairness and stability for competing DASH sessions using service endpoint functionality.

RQ3: How to choose appropriate quality levels for a DASH session.

The main contributions from my research are:

C1: A method for controlling the quality levels of DASH in the home gateway.
C2: A method for improving fairness among competing DASH sessions.
C3: A method for shaping traffic aggregates on access links with DASH components.
C4: A method for estimating available bandwidth on access links when DASH sessions are present.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for partial fulfilment of the requirements for the degree of philosophiae doctor.

This doctoral work has been performed at the Department of Telematics, NTNU, Trondheim, with Professor Poul E. Heegaard as main supervisor and with co-supervisor Professor Bjarne E. Helvik.

The majority of this work was conducted within the R2D2 Networks research project headed by UNINETT and SINTEF. This project was part of the European EUREKA/CELTIC program and financed through the Norwegian Research Councils VERDIKT program.

Several people have directly or indirectly contributed to the work presented in this thesis. First of all I would like to thank Poul E. Heegaard and Bjarne E. Helvik for all help and support over the last four years. In addition, it has been very valuable to discuss with Professor Yuming Jiang and through this improve my research. I would also like to mention my fellow Phd candidate Eirik Larsen Følstad. Although our research areas are not directly related, I have learned a lot from Eirik in terms of approach and methodology.

Through the R2D2 project I found inspiration and support in working with project partners and associated companies. The sincere interest and help from TV2, represented by Tor-Einar Eriksen was very important for me. He provided me with access to the knowledge and experience in TV2 with regards to state-of-the-art video streaming. TV2 also connected me with their technology partner, Vimond Technologies. The help and support from Vimond Technologies represented by Anders Instefjord was very valuable, and it gave basis for some of the published papers.

For me to pursue a PhD degree had not been possible without the support of my previous employer, NextGenTel. A special thanks to the former CEO of NextGenTel - Jan Dagfinn Midtun and my managers in NextGenTel - Kari Marvik and Erik Hovstad. By supporting me and granting me a part-time position during my PhD work they made it possible for me to do this from a financial perspective.

Finally, a special thanks to my mother, Bodil Villa. She gave me the early motivation and interest for higher education, and even when her son decided to pursue a PhD at the age of 40 she was very positive and proud.

Contents

Abbreviations

ANSI	American National Standards Institute
API	Application Programming Interface
BB	Bandwidth Broker
CBR	Constant Bit Rate
CDN	Content Delivery Network
CWND	Congestion Window
CLI	Command Line Interface
COTS	Commercial Off The Shelf
DASH	Dynamic Adaptive Streaming over HTTP
DiffServ	Differentiated Services
DSL	Digital Subscriber Line
DLNA	Digital Living Network Alliance
FEC	Forward Error Correction
FTTH	Fiber To The Home
GSM	Global System for Mobile Communication
HLS	HTTP Live Streaming
IntSer	Integrated Services
IETF	Internet Engineering Task Force
IPC	Inter Process Communication
ISO	International Organization for Standardization
ISDN	Integrated Services Digital Network
ITU	International Telecommunication Union
KBS	Knowledge Based Systems
MPEG	Moving Pictures Expert Group
NTNU	Norwegian University of Science and Technology
NIC	Network Interface Card
OECD	Organisation for Economic Co-operation and Development
OTT	Over The Top
PASTA	Poisson Arrival See Time Averages
P2P	Peer to Peer
PRM	Probe Rate Model
PGM	Probe Gap Model
QoS	Quality of Service
QoE	Quality of Experience
RFC	Request For Comments
RWND	Receive Window
RTP	Real-time Transport Protocol
RTSP	Real-time Streaming Protocol
RTCP	Real-time Transport Control Protocol
SDN	Software Defined Network
SNR	Signal to Noise Ratio
TCP	Transport Control Protocol
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
WFQ	Weighted Fair Queueing

Part I: Thesis Introduction

1. Introduction

Over the last 10 years - Internet access, and the use of Internet based services has experienced a growth which exceeds many of the earlier made predictions. It has evolved from something which was primarily used at universities and in the advanced business sector, to become a true commodity – also in the residential area. In many countries the maturity of this market is so that one thinks of broadband Internet access in the same way as one thinks of water and electricity. In other words, it has become one of the basic elements of our lives.

The drivers and motivation for the involved parties in this evolution has primarily been commercially based. New providers have seen this as an opportunity to become the "next generation telecom" provider, while existing ones have considered it as a necessary or strategic evolution of their legacy business.

Alongside with this, the Internet community in general has gained benefit from the evolution in the sense that new types of users have been introduced to the Internet, and this has contributed to a strong creative drive in terms of developing new services and applications for the Internet. The end users which have gotten used to have access to Internet are maturing in the sense that they are willing to use new Internet based services and are also willing to pay for them. Examples of this include emerging on-demand content delivery from the traditional TV broadcasting companies which now have as part of their core business to also deliver content over Internet. There are also indications on that more and more of traditional printed media are migrating over to digital on-demand delivery (newspapers, books etc).

On the terminal side the equipment vendors are also putting effort into facilitating easy and flexible ways of consuming Internet content. Small laptops (netbooks) and tablet PC's are growing in both amount and popularity in households, enabling users to access any service at any time. With this growing maturity among the end users, and also flexibility in terms of terminal types one could think of the period we are now in as the "age of Internet for devices and content consumption".

When end users are consuming more and more Internet based content and an increasing amount of this no longer is free, they will set their expectations higher concerning quality. The aspect of quality can be discussed from different perspectives, e.g. from the provider side or from the end user side. Traditionally, from the provider side the term QoS (Quality of Service) has been used to describe both the requirements and the actual performance of specific services using technical metrics (e.g. bitrate, delay, packet loss). As such, the approach has been very network and equipment centric and to some extent neglecting the end user dimension.

As the importance of end user acceptance and appreciation was acknowledged, a richer view on QoS emerged. The concept of QoE (Quality of Experience) term appeared and was defined to cover both technical and non-technical aspects. Thus, the traditional QoS definition was considered as a subset of QoE. The non-technical aspects added by the QoE approach were things like end user acceptability, perception and subjectivity.

Service providers have acknowledged that both QoS and QoE metrics are decisive for whether a new service becomes a success or not. Some providers even use the non-technical metrics from the QoE domain to compensate for shortcomings in the QoS domain. An example of this would be the availability of content on PC's and tablet which earlier were reserved for TV sets at very specific quality (bitrate) levels. The convenience of being able to watch a movie or similar content on whatever device you like, rather than a specific one - is clearly appreciated by users. Aspects such as convenience and flexibility with regards to how services are used are not covered by traditional QoS metrics, but could rather be considered as part of the QoE domain. The use of adaptive capabilities within services is also an example of the new way of thinking when engineering advanced services for the Internet.

There has been on-going research in the area of adaptive networks and services the last decade in the telecom industry in general – and in the Internet community in particular. Examples of this include the work [1] from 2004, in which the requirements for mobile devices to receive content and services adapted to criteria's such as user preferences, device type, bandwidth and location are discussed. Further on, in the work [2] from 2006, context aware provisioning for advanced Internet services is discussed.

The research efforts in this domain have been motivated by different things. On the network side, an example would be the capability to e.g. adapt routing paths to handle link or node failures in order to maintain end-to-end connectivity. On the service side, the focus on adaptive behaviour has primarily been driven by the Internet community for the purpose of facilitating efficient bandwidth utilization and making new services available.

Being able to adapt service requirements (e.g. bandwidth) both before and during service delivery is very appealing when using the best effort Internet traffic class for transport. For video services, or services where video is a component this capability has proven to be very effective. Following the early proprietary solutions, relevant standardization bodies (MPEG Forum, IETF) have recently published applicable standards in this domain for Dynamic Adaptive Streaming over HTTP (DASH) [3]. The topic is also being addressed in the current and future European research programs [4].

When delivering a service across the Internet to a specific user, it could be that many networks are involved and the type of user access could be of different types (fixed or wireless, DSL or Fiber) with unknown capacity. For a service provider with no relationship or agreement with the involved network operators this represents a very unpredictable transport infrastructure for their services. Despite this, a growing amount of providers are delivering services this way [5], and it is commonly known as the Over-The-Top (OTT) service delivery model. The use of adaptive services is a quite common approach in this domain in order to deal with the uncertainties of the involved transport infrastructure.

1.1 Problem Outline

The motivation for the research topic of this thesis is to contribute with improvements in the way services with video components are delivered across the Internet to end users. The ability to use best-effort Internet transport for advanced services is the fastest way to reach a high number of users on a global basis. Utilizing functionality in the end-points (i.e. client and server) to achieve this, rather than relying on implementation of QoS mechanism in the involved networks are appealing for many reasons. The use of QoS mechanisms would increase the cost of providing a service, and it would also increase the complexity in the involved networks. The more complexity you add to a network, the more challenging it also becomes to scale it while maintaining the same degree of service. Thus, whatever can be done in the end-points in order to support the delivery of advanced Internet services is considered an interesting research topic.

Consumer Internet traffic is already dominated by video services and forecasts indicate that by 2015 it will represent more than 60% of the total traffic [6] [7]. If all services are included which has video as a component, it is predicted to represent about 90% of the traffic. Thus, methods for optimizing the quality of video services and components are quite important.

Further on, in mature broadband markets the national Internet backbone infrastructure is dimensioned for zero congestion and thereby it provides high quality transport even for the best effort Internet traffic class. In these cases, it is also commonly seen that the OTT service delivery platforms are highly distributed using CDN nodes (cf. Section 2.2). These factors lead to that the main bottleneck for end users accessing demanding services such as video streaming, is the broadband access link which is a shared resource for all devices connected to a home network.

This leads to an interesting question with regards to which role the home gateway, as a transit node for all traffic in and out of a home network should undertake. In an OTT service delivery model both the home gateway and the end user clients can take an active part in enhancing service quality aspects, as they normally reside outside of the control domain for the network operators. This provides motivation for two of the research questions in this thesis (RQ1 and RQ3, cf. Section 1.3), which relates to autonomous performance control of DASH services delivered to a home network and how the clients can choose the appropriate quality levels for each service session.

For a user group present in a home network environment it is also interesting to investigate methods for improving inter-session fairness and session stability. The rationale for this is not only based technical aspects, but also a potential end user awareness of differences in experienced service quality. This is addressed by one of the research questions in this thesis (RQ2, cf. Section 1.3), which addresses the potential of achieving such effects using only service endpoint functionality.

1.2 Research Context

The research has been conducted as part of the Road to media-aware user-Dependant self-aDaptive NETWORKS – (R2D2 Networks) project. This project was funded by The Research Council of Norway. This project had as scope to work on techniques and tools to model, analyse and evaluate service usage in order to help service providers deliver according to agreed QoS and QoE requirements.

The project gave me freedom to define my own directions and focus for my work, but at the same time – support was given by project partners when needed. The close dialogue with project partners TV2 (Norwegian commercial TV broadcaster) and NTE (Norwegian regional network operator) was quite useful and motivating. TV2 was one of the early users of DASH technology as part of their Internet based services.

TV broadcasters such as TV2 do not normally have their own ISP operation or provide broadband Internet access to their customers. Thus, they are in favour of methods which enable them to provide services according to the OTT model. With both a strong interest from project partners in methods applicable for OTT service delivery, and also the obvious challenges in this domain – the focus of the research became OTT oriented.

Further on, NTE as the involved network operator was using FTTH technology to serve their customers. This provided an encouragement to first of all consider home networks served by a fixed broadband access in the research.

1.3 Research Questions

The novelty of the research approach and research questions defined in this thesis lies in the strict focus on methods which can be applied at service end-points (client, home gateway and CDN). This makes the research results of special interest for OTT service delivery, and thereby differentiates the thesis contributions from much of the related research efforts e.g. driven as part of the European Commission research programs. The projects financed under these programs normally include major network operators as participants, and thereby the focus is set accordingly.

For this thesis, the following research questions were defined – which reflect the focus on DASH services (due to popularity and high technical requirements), home networks connected to the Internet by a broadband access and the Over-The-Top service delivery model. The questions are all defined based on the assumption that functionality inside the network operator domain is not available due to either lack of interest from the network operator or absence of a joint business model involving all parties.

RQ1: In a home network how to (autonomously) control the performance of DASH based services.

In mature broadband markets and developed countries, the use of Internet from home is quite common. The amount of devices connected at home is also growing. In the early days of Internet a family would typically have a single PC, while now a mixture of several PC's, laptops, tablets and smart phones are all connected to the home network. Further on, as Internet services with video components are becoming more popular it is likely that multiple DASH based services will be delivered to a home network at the same time. This scenario changes to some extent the context for which DASH was originally developed. The case where multiple adaptive services are adapting to each other could create unfortunate situations. It is therefore interesting to lift this control challenge up to the level of the home network, and study whether it could be addressed there.

RQ2: How to provide fairness and stability for competing DASH sessions using service endpoint functionality.

The issue of fair treatment of services when there is a shortage of a certain resource (e.g. bandwidth) is relevant in many contexts. For DASH based services this is of interest as one could envision cases where sessions which were started earlier than others get the opportunity to reach their maximum quality level and to remain at this level. The later started DASH sessions across the shared home network access could then potentially be forced to remain at one of their lower quality levels. Assuming all users are equally important, or at least that the order of their DASH session start does not reflect a priority – such situations is not considered fair. Further on, there is also a chance that competing DASH sessions could make each other start fluctuating between quality levels. Although this could give an improved fairness over some time interval, it could represent noticeable service degradation for the users. This provides motivation for investigating new methods for improving fairness, while retaining stability for competing DASH sessions.

RQ3: How to choose appropriate quality levels for a DASH session.

For a specific DASH session, the controlling logic in terms of choosing quality level to be delivered from the server resides on the client side. This logic is not subject to standardization and would in this regard represent one of the sources for differences between commercial DASH solutions. As the choice of quality level leads to different traffic bitrates toward the client, and these changes appears on a semi-continuous basis, the client either directly or indirectly relates to how much available bandwidth there is for his session. The indirect approach to this would be to apply algorithms utilizing information from the client receive buffer such as filling degree and arrival rate. The direct approach to available bandwidth estimation would be to perform either active or passive measurements. For OTT service delivery, passive measurements are not easily achieved due to lack of access to the network itself. However, performing active measurements by means of probing is feasible.

1.4 Contributions

The following contributions represent the results of the research related to answering the research questions.

C1: A method for controlling the quality levels of DASH in the home gateway.

The suggested method is based on providing the home gateway which connects a number of users to the Internet with information about current DASH sessions. The information gives the home gateway knowledge about how many sessions are active, which quality level is currently chosen by each clients and which other quality levels are available for each session. The information is sent to the home gateway at regular intervals, or whenever changes occur. Based on this information, I have shown that the home gateway can act like a bandwidth broker in a more efficient way.

C2: A method for improving fairness among competing DASH sessions.

The suggested method for improving fairness among competing DASH sessions is based on making competing sessions different with regard to the quality level request interval used. I have shown that a mix of different quality level request intervals is better in terms of fairness among competing sessions, rather than a default fixed and equal interval. The assignment of request interval to be used can be done by a stochastic process or some other algorithm.

C3: A method for shaping traffic aggregates on access links with DASH components.

The suggested method for achieving a shaping effect for traffic aggregates is based on the same principle as for C2, but the assignment of request intervals are not done by a stochastic process. It assumes that competing sessions are delivered from a Content Delivery Network (CDN) node which is able to identify a group of session's relationship to a specific home network. Based on this grouping the CDN assigns fixed and unique request intervals to each of the competing sessions. The method only requires additional logic on the CDN side.

C4: A method for estimating available bandwidth on access links when DASH sessions are present.

The suggested method is addressing in particular the scenario where cross-traffic on an access link is very bursty. It utilizes the presence of periodic traffic patterns in the cross-traffic for aligning of active probing according to strata. The idea is that in order to maximize the information obtained from of a sequence of probe packets, it is best to send them during the traffic bursts of the cross-traffic, and not in the more silent periods. In addition, the amount of probe traffic must be kept as low as possible in order to minimize the effect on the cross-traffic. The method can be considered as a supplement to other more generic approaches, as it mainly represents value if there are periodic patterns in the cross-traffic.

1.5 Papers

For all the papers included in this thesis, except for P5 and P11, the initial idea was mine and I did all the work related to the research itself and writing the paper.

For paper P5, I had the original idea for the research topic investigated, defined the experiments from a technical perspective and did all the analytical work. The implementation of the network solution used in the experiments was done by me, in co-operation with Anders Instefjord which provided encoding services for the video material to be used.

For paper P11, I had the original idea for the research topic investigated, and defined the experiments from a technical perspective. The implementation of the network solution used in the experiments was done by me, in co-operation with Anders Instefjord which provided encoding services for the video material to be used. I was responsible for recruiting users to take part in the experiment, and co-operated with Katrien De Moor in order to execute the experiments. The preparation of questionnaires, processing of responses and in-depth analysis of the findings was done by Katrien De Moor.

The contributions from my advisor, Poul E. Heegaard, were composed of useful comments and discussions related to choice of method, analysis and presentation of the results. He also reviewed all papers before submission.

P1 Bjørn J. Villa and Poul E. Heegaard: "***Monitoring and Control of QoE in Media Streams using the Click Software Router***". Published in Proceedings of Norsk Infomatikkkonferanse 2010 (NIK 2010) Tapir Akademiske Forlag, ISBN 978-82-519-2702-4. pp. 24-33 Nov 22-24, 2010, Gjøvik, Norway
Relevance to this thesis: This paper is based on my initial study of DASH services. In this work I studied the possibility of influencing the quality level selection done by a DASH client using selective intercept and postpone TCP ACK packets (PostACK). This method was implemented in an experimental home gateway router using the Click Modular Router software solution, and through this it was shown that PostACK can be used to control DASH quality levels on a per session basis. The paper is a partial answer to RQ1 and contributes towards contribution C1.

P2 Bjørn J. Villa and Poul E. Heegaard: "***Towards Knowledge-driven QoE Optimization in Home Gateways***". Published in Proceedings of The Seventh International Conference on Networking and Services 2011 (ICNS 2011), ThinkMind Digital Library, ISBN 978-1-61208-133-5, pp. 252-256, May 22-27, 2011, Venice, Italy.
Relevance to this thesis: This paper focuses on how a home gateway can use knowledge about DASH services in order to increase the average quality level achieved when using knowledge based bandwidth broker schemes. The schemes are investigated by means of simulations. The paper is a partial answer to RQ1 and contributes towards contribution C1.

P3 Bjørn J. Villa and Poul E. Heegaard: "***A Monitor Plane Component for Adaptive Video Streaming***". Published in Proceedings of Norsk Informatikkonferanse 2011 (NIK 2011) Tapir Akademiske Forlag, ISBN 978-82-519-2843-4. pp. 145-154, Nov. 21-23, 2011, Tromsø, Norway.
Relevance to this thesis: This paper focuses on the feasibility of providing a home gateway with knowledge about DASH type of services, and describes an experimental implementation of a DASH client supporting this. In this regard, the paper supports paper P2. The paper is a partial answer to RQ1 and contributes towards contribution C1.

P4 Bjørn J. Villa and Poul E. Heegaard: "***Improving Perceived Fairness and QoE for Adaptive Video Streams***". Published in Proceedings of The Eighth International Conference on Networking and Services, 2012 (ICNS 2012) Thinkmind 2012, ISBN 978-1-61208-186-1. pp. 149-158, March 25-30, 2012, St. Maarten, Netherlands.
Relevance to this thesis: This paper investigates the possibility of achieving improved fairness among competing DASH session without sacrificing session stability. It discusses different fairness measures in both QoS and QoE domain, and suggests using session stability as an indicator on perceived fairness. The method investigated is based on changing the rate request intervals from a default fixed and equal value for all sessions over to a stochastic parameter of either the uniform or negexp type. The paper is a partial answer to RQ2 and contributes towards contribution C2.

P5 Bjørn J. Villa, Poul E. Heegaard, Anders Instefjord: "***Improving Fairness for Adaptive HTTP Video Streaming***". Published in Proceedings of 18th EUNICE/ IFIP WG 6.2, 6.6 International Conference 2012 (EUNICE 2012) Springer, ISBN 978-3-642-32807-7. pp. 183-193. Aug. 29-31, 2012, Budapest, Hungary.
Relevance to this thesis: This paper takes the analysis of the method suggested in P4 further, by means of an experimental study. However, the experiments are focused on using per session unique segment request intervals rather than random intervals. The main reason for this difference in approach is that the random interval selection is quite difficult to achieve in real life, as each interval corresponds to specific encoded content. The paper is a partial answer to RQ2 and contributes towards contribution C2.

P6 Bjørn J. Villa and Poul E. Heegaard: "***Improving Fairness in QoS and QoE domains for Adaptive Video Streaming***". Published in The International Journal On Advances in Networks and Services, Volume 5 n. 3&4, 2012, http://www.iariajournals.org/networks_and_services/, IARIA, ISSN: 1942-2644. pp. 291-303.
Relevance to this thesis: This invited journal paper is based on an alignment of the work in P4 and P5, and a combined analysis of the results. The paper is a partial answer to RQ2 and contributes towards contribution C2.

P7 Bjørn J. Villa and Poul E. Heegaard: "***Group Based Traffic Shaping for Adaptive HTTP Video Streaming***". Proceedings of the 27th IEEE International

Conference on Advanced Information Networking and Applications (AINA 2013). IEEE Computer Society Press, ISBN 978-0-7695-4953-8. pp. 830-837. March. 25-28, 2013, Barcelona, Spain
Relevance to this thesis: This paper analyses whether the method of using per session unique segment request intervals has a shaping effect on the traffic aggregate for a group of competing DASH sessions. The purpose of this study was to better understand the fairness effects shown in P4-P6. The paper supports the findings of P4-P6 in terms of answering RQ2 and thereby also contributes towards contribution C2. It also represents contribution C3 by itself.

P8 Bjørn J. Villa and Poul E. Heegaard: "***Detecting Period and Burst Durations in Video Streaming by means of Active Probing***". Presented at The 3rd International Conference on Computer Communication and Management (ICCCM 2013) May. 19-20, 2013, Copenhagen, Denmark. Published in International Journal of Computer and Communication Engineering ISSN 2010-3743, volume 2 (4), 2013, pp. 460-467.
Relevance to this thesis: This paper presents a method for detecting periodic behaviour and burst duration by means of active probing. The work is motivated by an objective to establish a traffic profile for cross traffic on an access link, which can further be used for available bandwidth estimation. The paper is a partial answer to RQ3 and contributes towards contribution C4.

P9 Bjørn J. Villa and Poul E. Heegaard: "***A Measurement Study of Active Probing on Access Links***". Published in Proceedings of the 19th EUNICE Workshop on Advances in Communication Networking 2013 (EUNICE 2013) Springer, ISBN 978-3-642-40551-8. pp. 124-135. Aug. 28-30, 2013, Chemnitz, Germany
Relevance to this thesis: This paper analyses two different approaches for active probing, i.e. packet pair probing and one-way-delay probing. By means of experiments it is shown how these methods differ in their ability to detect cross-traffic amount. The paper is a partial answer to RQ3 and contributes towards contribution C4.

P10 Bjørn J. Villa and Poul E. Heegaard: "***Estimating Available Bandwidth on Access Links by Means of Stratified Probing***". Presented at The 6th International Conference on Computer and Electrical Engineering (ICCEE 2013), October. 12-13, 2013, Paris, France. Published in International Journal of Electrical Energy, ISSN 2301-3656, volume 1 (4), 2013, pp. 213-221
Relevance to this thesis: This paper analyses the capabilities of a suggested method for active probing and estimation of available bandwidth. It is based on the results from P8. The paper is a partial answer to RQ3 and contributes towards contribution C4.

P11 Bjørn J. Villa, Katrien De Moor, Poul E. Heegaard, Anders Instefjord: "***Investigating Quality of Experience in the context of adaptive video streaming: findings from an experimental user study***". Published in Proceedings of Norsk Infomatikkkonferanse 2013 (NIK 2013) Tapir

Akademiske Forlag, ISBN 978-82-321-0365-2 . pp. 122-133,Nov 22-24, 2013, Stavanger, Norway
Relevance to this thesis: This paper is based on an experimental user study with the objective of investigating how end users perceives the quality of DASH based streaming using different profiles and usage scenarios. The paper provides input to both RQ1 and RQ2, but mainly contributes towards contribution C2.

1.6 Relations

The relations between papers and contributions are described in the previous section. In order to describe how it relates to my research questions, the following visualization should be considered (cf. Figure 1). As the numbering of the papers is done according to research timeline, it is clear that the research questions RQ1-RQ3 also have been addressed sequentially. The bi-directional arrow between C2 and C3 indicates that although these are separate contributions, they are closely connected due to similarity in underlying method.

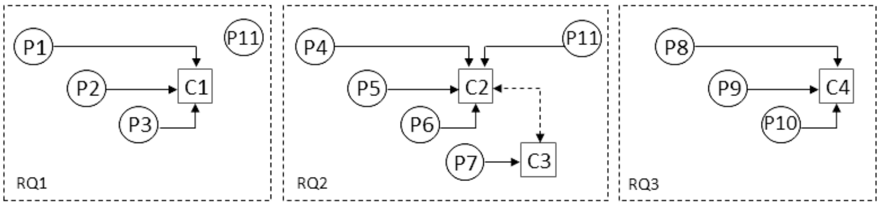


Figure 1: Relations between papers and research design / model

The final paper included in the thesis (P11) connects back to the two first research questions, and contributes to C2 as part of the intention was to re-visit and investigate a hypothesis presented in P4 regarding perceived fairness.

The composition of findings in each of the papers and how they address their respective research questions will be described in more detail in the results section.

1.7 Thesis Structure

The structure of this thesis is as follows. In Chapter 2 an overview of State of the Art is given. In Chapter 3 a description of my research design is presented, with focus on the research goal, model and method. In Chapter 4 the research results are presented in terms of answers to research questions. In Chapter 5 the thesis contributions are evaluated against the current view of related research and validity. In Chapter 6 the thesis conclusions are given.

2. State of the Art

In each of the included papers, there is a section on related work for the specific topic investigated and what was considered as state of the art at the time of writing. In the evaluation section of the thesis an updated view of the relevance for the research questions and contributions are given. Thus, in this section - focus is put on state of the art for the main technology areas of relevance for the thesis research on a somewhat higher level.

The main technology areas of interest which will be presented in the following sections are as follows, with relevance for the thesis work indicated.

- **Service Quality**: Provides a baseline for the performance related parts of the research topics, and in particular the motivation for bridging QoS and QoE metrics. (cf. papers: P2, P4, P6 and P11).
- **Service Delivery Architectures**: Provides insight into the evolution of service delivery architectures for demanding services such as video. It applies to the thesis research as a whole due to its significance for OTT providers, but in particular it motivates research on methods which can be applied on the service end points only - such as CDN nodes (cf. papers: P4, P5 and P7).
- **Home- and Access Network Control**: Provides an overview of related research on mechanisms for home- and access network control. The relevance for the thesis work is the role and evolution of home gateways, and in particular client interactions with the home gateway for the purpose of controlling service quality (cf. papers: P1 and P3).
- **Content Encoding**: Provides an overview of the evolution in video encoding and positions the DASH concept in relation to this. The relevance for the thesis work is to make it clear that the contributions of the thesis work can be applied independent of the specific video encoding technique used (cf. papers: P1- P11).
- **Adaptive Services**: Provides an introduction to the use of adaptive behaviour for key services (access, voice and video). The relevance for the thesis work is to provide justification for focusing on this type of service behaviour, and also the required understanding of the DASH concept in particular (cf. papers: P1- P11).

- **Transport Protocols**: Provides the status for TCP protocol versions used in major OS version. The relevance for the thesis work is that although DASH services are based on TCP, the interactions between TCP layer mechanisms and DASH application layer mechanisms are not fully understood. This was the basis for investigating the potential of using TCP mechanisms to control DASH quality levels in a resource efficient way (cf. paper: P1).

2.1 Service Quality

The concept of service quality is defined and understood from many perspectives and has evolved over time. Before the dawn of the Internet it was a very close relationship between the network provider and service provider roles, and the degree of service integration was moderate. The introduction of Internet also increased the level of competitiveness in the market, by making the world a single market. As the early definitions of service quality were made before the Internet became commodity, they should therefore be understood accordingly. These early definitions were made using metrics from the network domain and gave origin to the frequently used term Quality of Service (QoS). In the early specifications from standardization bodies such as ITU-T, the focus was on the telephony service and target values for QoS metrics relevant for this in particular [8]. Today, the commonly used QoS metrics (delay, packet loss, jitter) are more general and applies to a whole range of packet switched services and combinations of such.

2.1.1 Internet QoS Architectures

The Internet as originally defined was a pure best-effort network where all packets where considered equal. The aspects of delay, packet loss and jitter were not of any concern at this time. However, this is not entirely true as the use of different transport protocols (UDP, TCP) started quite early. By mapping different services to either TCP or UDP one would get a basic difference in terms of traffic handling, and this was started around 1980. One might say that the choice of UDP as transport protocol reflected a scenario where “drop rather than delay” was preferred by the application, while the choice of TCP reflected a “delay rather than drop” preference. Even though we do not think of this reliable data transfer capability of TCP today as a QoS differentiation mechanism, it could indeed serve as such.

In the Internet community there are two main architectures for implementing QoS and they have very different approaches. The first one is called Integrated Services (IntServ) [9], while the other is called Differentiated Services (DiffServ) [10]. The main difference between these two models is that while IntServ has an end-to-end control approach, DiffServ has a hop-by-hop approach. At this level one could say that IntServ is the typical telco way of thinking, while DiffServ is more the Internet way of thinking. However, both models have been subject to work within the IETF and have resulted in numerous RFC’s. As such, both models are candidate Internet QoS architectures and are in use today.

Common for both the IntServ and DiffServ model is that the network must be able to handle traffic (i.e. packets and flows) different based on their relation to traffic classes. These traffic classes have target QoS metrics associated, which reflects the requirements from services using them.

Availability of IntServ or DiffServ functionality in networks which are part of the Internet today cannot be assumed, and even if it should be available in some parts for the Internet – the potential absence of it end-to-end would be a challenge. Further on, using this type of functionality comes at a cost e.g. for an OTT provider. Thus, both the availability and cost aspects of utilizing Internet QoS architectures are reasons for OTT providers to consider other options.

2.1.2 Definitions of QoE

Despite the evolution of QoS metrics and their applicability to new services, they are not rich enough to cover all aspects of service quality. The main reason for this is that although a service can be engineered according to certain QoS metrics, it is never a guarantee that the actual user of the service is satisfied. User expectations are by nature different, and engineering for such diversities are quite challenging. This was also recognized by ITU-T as they made an amendment to their relevant specification in this field [11] and also a new family of relevant specifications in the E8xx-series emerged [12][13][14]. This amendment introduced the concept of Quality of Experience (QoE) and defined this as "The overall acceptability of an application or service, as perceived subjectively by the end user". This new definition was a radical change in the way service quality was understood and gave directions for the future in terms of how services should be made. One should of course be careful to give ITU-T all the credit for this definition. The fast growing Internet community is likely to have made significant contributions in this domain.

The initial definition of QoE as per the ITU specification [11] provided a significant broader view on what was important in terms of quality. The introduction of terms such as acceptability, perception and subjectivity was a big step away from network engineering and more towards social science. The success of new services (e.g. VoIP) on the Internet which earlier were believed to require QoS mechanisms, made the very conservative telecom industry realize that something was happening. When making a service available on other platforms and in new contexts, it was seen that users were willing to accept somewhat lower measureable quality on the service level. Looking even further back in time, the same phenomena took place when mobile telephony was introduced. The voice service had a lower quality, but was provided with flexibility in terms of mobility. Thus, the mobile telephony service was perceived as both acceptable and attractive.

Despite the already rich definition of QoE by ITU there has been an evolution also in this field. The original definition still holds, but it has been criticized due to maybe a too narrow interpretation [15]. Recently a new definition of QoE as "the degree of delight or annoyance of the user of an application or service" was proposed [16]. This definition intends to even include how the end users expectations, personality and state of mind influence their satisfaction.

Taking the complexity in this field even further, a whole range of human, system and context related factors may influence the end user QoE. This implies that a QoE oriented approach for Internet service delivery requires a more holistic and interdisciplinary approach [15].

In this thesis work a suggestion is made regarding a new QoE metric. The metric is called perceived fairness, and a hypothesis is made regarding the relationship between this and the stability of DASH sessions (cf. paper: P4). This idea behind this hypothesis was that users may consider stability issues to be caused by others users in e.g. the same home network. In the final paper included in this thesis (cf. paper: P11) the validity of this hypothesis is attempted verified by performing experiments with users watching videos of variable quality profiles in different social settings (alone vs. groups).

2.2 Service Delivery Architectures

A service delivery architecture in the context of this thesis primarily describes how a service is technically implemented when delivered across the Internet. It could also describe aspects of the business model as it indicates relations and interactions between different actors regulated by agreements (e.g. Service Level Agreements) including quality attributes.

From a service perspective the Internet can be viewed as one network, although it in reality consists of thousands of different autonomous networks working together. The view on Internet as a single global network is fascinating from a service perspective as it would facilitate any service provider to reach any customer around the world. For many services this is actually how it works, but for other – more demanding services – it is not that straightforward. Basic services do not expect much from the Internet other than end-to-end connectivity with an elastic capacity, and this is the fundamental service provided by the Internet. Services residing in this domain have never been a challenge, and will most likely continue to be served satisfactory. The more demanding services, those which have certain expectations regarding QoS are a completely different story. The challenge in this scenario would be the lack of QoS guarantees end-to-end on the Internet. Even though mechanisms exist, making them work across different networks in a similar way and at affordable prices still remains to happen.

In the early days of delivering demanding services (e.g. video) across the Internet, the end user expectations and requirements regarding quality is assumed to have been rather low. This assumption is based on the theory behind the Bell Curve [17] which describes the typical users of a new product entering the market. The first user groups are stated as the innovators and early adopters. Such users are characterized by curiosity and willingness to embrace new things. Following the growth of the Internet in terms of capacity and amount of users connected, the characteristics of the users also changed. The users were no longer early adopters, but instead early and late majority users as per the Bell Curve. These users are recognized as more careful and somewhat sceptical. Thus, it is reasonable to assume that both expectations and requirements with regards to quality increased. In other words, the successful introduction of demanding services on the Internet gave birth to the interesting challenge of not only producing more service units but also with a higher quality.

The solution to maintain or even improve the quality of services on the Internet as they grew in popularity was the introduction of acceleration functions. As the most popular service on the Internet in the early days was basic web access, efforts were made to accelerate this in particular. The acceleration was done by bringing popular content closer to the users in a dynamic way, driven by what the users actually were requesting. The method used in order to achieve this was to intercept the http traffic at certain points in the networks where the accelerators were located and see whether the requested content was available in the local cache. This concept worked quite well for a period until http intercept became too processing intensive, and also new services appeared which did not work when being subject for this type of intercept (e.g. web banking). Even though the technology has changed, web accelerators are today recognized as the first generation of Content Delivery Networks (CDN).

Today, CDN still is based on the same principle of bringing popular content closer to the customers in a dynamic way but the technical solution is changed. One important change is that instead of http intercept which was used in the web accelerators, the CDNs of today are based on using http redirects. Using http redirect instead of http intercept allows for the CDN nodes to no longer be a transit node for regular traffic. Instead, they serve only as source for the specific content stored on them whenever users are directed towards them.

This gives a significant benefit in the regard that non-accelerated service are not touched by the CDN and thereby also not potentially damaged. Although the CDN concept can be used for most services which are fetching content from a remote location, it is primarily used for http based services.

Another major difference between the early web accelerators and the CDN's of today is that when deployed, they no longer are dedicated to a single user or company. Deploying shared CDN's has become a service in itself on the Internet and among the pioneers in this field you would find Akamai Technologies [18]. Based on corporate statements from Akamai, between 15-30% of all web traffic on the Internet is delivered from their CDN.

The deployment of CDN represents value for both the network operator and the end user. For the network operator it will reduce the traffic load both internally in his network and also on the external links towards the global Internet. As such, it normally represents a cost saving for the operator. However, the operators still tend to be somewhat annoyed that they are not normally able to increase their revenue by allowing CDN providers into their network. The value of CDN for end users is related to both capacity and delay parameters. As the communication path between the end users and the server side (CDN) becomes shorter, there are less network elements involved and therefore a more predictable capacity situation. Normally, this would imply that the main congestion point for the user is the access link. The capacity on the access link is under control by the end users as it would be per his access subscription. A shorter communication path also contributes to reduced delay in the communication, something which is especially important for interactive services. In studies [19] done by Google it has been shown that the response time experienced when performing web search has an impact on how frequent the users are actually performing search operations. In another work [20] it is shown that users are very sensitive to startup delays when watching online video, especially when they attempted to watch video clips of an expected short duration. This would apply to a lot of the content provided by e.g. YouTube.

The next generation of CDN networks may utilize the potential which lies in peer-to-peer (P2P) technology. The P2P concept is based on a decentralised and distributed architecture in which nodes are equal in the sense that they all act as both suppliers and consumers of content. This concept has been popular in non-commercial Internet communities which share electronic media of different types (e.g. music) between themselves for many years. However, the concept can also be used in other scenarios and in the work [21] a hybrid CDN-P2P architectures is presented. In this work suggestions are made with regard to how the streaming capacity of P2P nodes can be used to supplement the capacity of the CDN itself. This is done by gradually introducing clients as P2P nodes for the purpose of onwards distribution of specific content, based on what they are requesting themselves from the CDN. Whenever enough P2P nodes are in operation for a specific media file, the CDN can do a handoff to P2P and stop serving the specific file.

Some guidelines for planning and dimensioning of such hybrid architectures are also presented. In [22] a hybrid CDN-P2P architecture is discussed for two of the leading CDN operators - Akamai and Limelight. The gains of a potential deployment of such a solution is analysed based on available traffic traces. Their findings indicate a significant potential in this area, in terms of being able to facilitate the continuous growth of Internet video in an economically feasible manner.

Focusing on the importance of CDN for Internet services with video components it is clear that it represents a very critical element. A highly distributed architecture for serving the growing amount of video consumption is indeed required, and the CDN concept has so far proven to be the right choice. However, in terms of utilizing the strengths of the network operator beyond just providing hosting for CDN servers remains to be investigated. Surprisingly little (if any at all) effort is put into finding ways which the network operator could contribute to CDN solutions in order to enhance the concept further. The view of a CDN as a Knowledge Based System with input from the network domain remains to be invested in more detail. An example of promising research seen in this domain is the work [23] where the use of agent technology is suggested as part of future CDN architectures. An agent component named AMonitor is suggested, and this performs various types of on-demand load monitoring which is used as input to the CDN platform. It also includes functionality for estimating network parameters along the path between clients and the CDN nodes.

In this thesis work new methods are suggested which can be applied in CDN platforms for the purpose of achieving improved fairness and also to achieve traffic shaping effects (cf. paper: P4, P5 and P7).

2.3 Home and Access Network Control

There has been a lot of research on Internet access solutions and home networks over the last 10 years. The main reason for this has been the rapid growth in broadband Internet and associated services. An important part of this research has been funded through the European Commission research programs. Within these programs, there are several interesting projects with interesting results in the domain of home and access network control. The main ones with relevance for this thesis work are the MUSE project [24], the ALPHA project [25] and the OMEGA project [26].

The MUSE project deserves special attention as it represents an important part of the foundation for this thesis research. The architecture described in the MUSE project deliverable DB1.8 - Advanced features for MM enabled access platform [27], introduced and discussed the roles of Monitor Plane and Knowledge Plane components for the purpose of enhancing QoE for Internet based services. The responsibility of the Monitor Plane was to acquire different types of information suitable for characterizing end-to-end connectivity and to present this to the Knowledge Plane. The methods used for information collection could be either passive or active measurements. As an example of a candidate method to be used the MUSE paper [28] presents a TCP monitoring algorithm named ANTMA (Access Network TCP Monitoring Algorithm). This algorithm is based on passive monitoring of transit IP packets at an intermediate node (e.g. the access node or home gateway). The output of ANTMA is estimations of upper and lower limits for packet loss, in both upstream and downstream direction. A more comprehensive approach tailored for multicast RTP-based services is described in another MUSE paper [29]. In this work, RTP/RTCP monitoring is used to extract information about packet loss, delay and jitter.

When the Knowledge Plane receives information from the Monitor Plane it would be processed for the purpose of anomaly detection, root cause determination, selection of candidate QoE restoring action and finally network optimization. An example of anomaly detected could be packet loss (as estimated by ANTMA in the Monitor Plane) which to some extent could be corrected by increasing the amount of Forward Error Correction Coding (FEC). The change in FEC would then represent the QoE restoring action according to the MUSE research.

The MUSE architecture was taken further in [30] [31] by including an Action Plane as the third component used in QoE optimization. In addition to this, a more generic Knowledge Plane architecture is presented using a neural network approach. The contribution of the neural network mechanism is to make the Knowledge Plane able choose or combine alternative optimization actions. The alternatives considered in the study are adjusting FEC and/or service bitrate. The approach of adjusting service bitrate in order to maintain or improve QoE is in line with the ideas behind DASH [32] based service delivery.

The MUSE project also presented interesting ideas on how home gateways could evolve into something more than just very basic connectivity devices. In [33] the requirements for future multi-service home gateways are discussed with focus on management aspects and operating systems used. The management part is expressed according to the needs of each actor around the gateway (home user, access provider, service provider) and also the technologies used for management agents. In light of the differences in this field, the paper presents the concept of virtualizing the home gateway into separate parts based on the Open Software Gateway Initiative (OSGi) technology [34].

Looking further into the home network and the wide range of clients present in this domain and the diversity in terms of services requested, it becomes interesting to follow the evolution of Universal Plug and Play (UPnP) support for QoS. The original intention of UPnP [35] was to simplify the connection of devices and services in a home network. Over the years this has evolved to also include support for QoS setup of traffic streams [36], which also includes an admission control component. By using the admission control component in UPnP, the QoS implementation gains some similarities with the IntServ QoS architecture.

In the ALPHA project [25] a mapping of UPnP QoS parameters over to a GMPLS domain is proposed, which could be relevant in the case where the access part is an Active Optical Network (AON). The use of GMPLS on this access type includes MPLS-TE for routing and RSVP-TE as resource reservation protocol. The contribution of the paper is quite interesting as is proposes a detailed parameter mapping between UPnP QoS and GMPLS/RSVP-TE QoS which could be used for inter-domain signaling. If implemented this could potentially provide strong end-to-end QoS guarantees. Even though it is stated in the paper that MPLS is being pushed toward the end-customers it remains to see if this really will take place. The use of MPLS in service provider networks today normally stops at network edge, but as customers are migrating to fiber based access solutions the proposed concept might become reality.

In the OMEGA project [26] the objective was to develop a user-friendly home access network capable of delivering high-bandwidth services and content. The use of UPnP QoS is considered as an important component also in their work. In the paper [37], Admission Control and Drop strategies for a UPnP-QoS controlled home network to guarantee QoS for multiple concurrent services is proposed. The motivation for the work is to improve the capabilities of the UPnP-QoS architecture to handle situation where the network becomes congested. The use of Admission Control on the Internet is in line with the ideas behind the IntServ QoS model [38], but its use on the Internet of today is limited. Although not a typical topic subject for standardization there are some initiatives also within the IETF. Some examples of this is the recent RFC 6601 [39] in which a Generic Connection Admission Control (GCAC) Algorithm Specification for IP/MPLS Networks is presented, and the RFC 5559 [40] which describes a Pre-Congestion Notification (PCN) Architecture.

The PCN architecture from RFC5559 is studied in [41] with regards to how well it performs when applied to video services. The PCN architecture is a measurement based approach, in the sense that instead of requiring detailed per flow traffic metrics it relates to the total bandwidth consumption. The initial work within the IETF on PCN addressed the case of admission control for inelastic flows with a limited degree of burstiness in the aggregated traffic pattern. When using TCP and DASH for video transport, the flows becomes elastic and with a more bursty traffic pattern. Through an experimental evaluation of PCN when applied to video services, the paper [41] presents interesting guidelines on how PCN can be modified and configured to protect video services. The modifications include an adaptive algorithm for configuring the PCN metering, a buffer mechanism which attempts to reduce the aggregate variability and also a video rate adaptation algorithm. The last part, i.e. the rate adaption algorithm suggested touches on a key part of DASH, and suggests changing this from a client driven mechanism to a network controlled mechanism. This approach makes DASH based services less suited for OTT delivery, and more appealing to the network operators.

In this thesis work an approach for controlling DASH quality levels by TCP ACK delay is described and verified through experiments in a testbed (cf. paper: P1). The idea behind this approach was to introduce as little additional processing in the home gateway as possible when controlling DASH sessions. The required client communication with the home gateway is also implemented using standard HTTP messages (cf. paper: P3) for a Microsoft Silverlight DASH solution.

2.4 Content Encoding

How content is encoded before transmission across the Internet towards the end users is important from many perspectives. From a network capacity and congestion perspective it is preferred that the resulting bitrates are as low as possible. However, from the end user and content provider perspective it is more important how the actual quality of the content is perceived. Clearly, as the bitrate used to some extent reflects the amount of information carried – it will also reflect on the quality. In order to get a deeper understanding of the quality one has to go beyond just bitrate and look at the actual techniques used for encoding.

The amount of information contained within an encoded video stream depends on many things. Aspects such as spatial resolution, colour bit depth, gamma, chroma subsampling, colour space, frame rate, audio bit depth and audio sampling rate are all important areas in which the encoding techniques may be different [42].

The relevant standardization bodies involved in developing standards for content encoding, and video encoding in particular are the Moving Picture Expert Group (MPEG) of ISO/IEC and the Video Coding Expert Group (VCEG) of ITU-T. Although they work side-by-side in many cases their focus is slightly different. MPEG specializes in content encoding for broadcast services, while VCEG focuses on the telecommunication domain (i.e. Internet).

Well known standards for video encoding are MPEG-1 [43]/H.261 [44], MPEG-2 [45]/H.262 [46], MPEG-4 [47]/H.263 [48], MPEG-4 AVC [49]/H.264 [50]. The most recent work in this area is covered by MPEG-H [51]/H.265 [52]. In the continuation of this section, the MPEG notations will be used.

The real growth of high quality video content on the Internet was made possible by the transition from MPEG-2 to MPEG-4 AVC. The differences in terms of engineering philosophy and algorithms applied in these two encoders are quite significant. Bandwidth savings by using MPEG-4 AVC rather than MPEG-2 varies to some extent depending on the content. However, bandwidth savings between 30-60% are quite common [53].

The next leap in capabilities for video encoding is represented by MPEG-H/H.265, also known as High Efficiency Video Encoding (HEVC). Some of the important aspects of this are the encoding of up to 8192x4320 resolutions, 12-bit colour depth, 4:4:4 chroma subsampling, up to 300fps and bitrates in the gigabit range. Relative bandwidth savings by using HEVC rather than MPEG-4 AVC is foreseen to be about 50%.

For the sake of clarity, the MPEG-DASH [3] specification does not introduce new methods for encoding but only media presentation descriptions and segment formats. As such, the DASH concept can be applied independent of which video encoding is used.

This thesis work does not relate to any specific content encoding scheme, but rather introduces mechanisms which can be applied to any DASH solution. This applies to the suggested method which enhances fairness and provides traffic shaping effect (cf. paper: P4 and P7), and also to the stratified probing approach for estimating available bandwidth (cf. paper: P8, P9 and P10).

2.5 Adaptive Services

The first enabler of adaptive services on the Internet was the introduction of TCP as transport protocol with its congestion avoidance and control mechanism [54]. The evolution of TCP will be discussed closer later in this thesis, while the focus in this section is adaptive functionality on other layers in the protocol model. The common aspect of such services is that they are able to change their behaviour or requirements according to certain events whenever they occur. The purpose of this change would be to either maintain a continuous mode of operation or to seek some kind of optimal operation. The applicability of this principle to both lower and higher levels of the Internet protocol model is outlined in the following.

2.5.1 Rate Adaptive Access

The basic service provided to Internet users is the access. This is based on a certain access technology and provides the users with an access capacity towards the Internet according to their subscription. Internet access started initially with dial-up solutions, but migrated over to fixed and always-on type of solutions as broadband Internet became a reality. The fast growing demand for broadband services required the network operators to utilize as much of the existing infrastructure as possible, in order to provide services according to the demand. Therefore, it was natural to use new generations of Digital Subscriber Line (DSL) technology on the already present telephony access network to provide broadband Internet access. The first generation DSL technology used for this purpose was Asymmetric DSL as per the ANSI specification [55] from 1998. Following this early specification from ANSI the ITU became more active and took over the majority of the standardization efforts up until today. According to OECD statistics [56], DSL technology is still dominating the market as it is being used on more than 50% of all broadband accesses in the world.

The DSL technology is sensitive to several types of noise which could lead to a degraded signal quality and in the worst case a complete loss of connectivity. In response to this challenge the ITU introduced as part of the ADSL2+ specification [57] the rate adaptive capability. The basic operation of this is that the modem part of the DSL circuit can dynamically adjust the upstream and downstream bandwidth according to the current Signal to Noise Ratio (SNR). There are several benefits of this mode of operation for both for the network operator and the customer. From the customer side it would automatically provide the maximum possible capacity on the local loop being used. From the network operator side it would obviously require less operation and maintenance.

2.5.2 Rate Adaptive Voice

The digital telephony service as provided by e.g. ISDN carried voice encoded according to the G.711 specification [58] from ITU published in 1972 with more recent revisions. This codec applies a moderate degree of compression (1:2) and gives a bi-directional capacity requirement of 64Kbps. Due to its simplicity in implementation (SW only) and also the fact that there is no licensing fee, it also became the first codec used in Voice over IP (VoIP) solutions. The use of G.711 to encode voice normally gives a very good quality, as perceived by end users. In addition to the G.711 codec there is a wide range of other codes as well with different characteristics and requirements, reflecting more specific usage scenarios. For digital mobile networks such as GSM where bandwidth is of concern, the first codecs used [59] [60] required only 6.5Kbps or 13Kbps. The resulting voice quality as perceived by users was lower, but still acceptable due to the flexibility offered by mobile terminals.

A more recent type of codec used in mobile networks for the purpose of increasing the voice quality is based on Adaptive Multi-Rate - Wideband (AMR-WB). Codecs based on this are standardized by both ITU-T [61] and 3GPP [62]. AMR-WB operates with nine different bitrate between 6.6Kbps and 23.85Kbps. Although the bitrates are still lower than the one used for G.711, it still gives an improved voice quality. The reason for this is that G.711 is based on narrowband sampling (300-3400Hz) while AMR-WB is based on wideband sampling (50-7000Hz). The selection of which bitrate to use for the codec is done based on both the quality of the current radio connection and the context of the voice session (speech, music, conferencing etc.). Mobile network operators providing services based on this technology uses the term “HD Voice” to describe their capabilities in this domain.

An alternative to have the codecs themselves implement adaptive behaviour, there is a possibility of having additional control algorithms in the respective voice application. Such algorithms could make the application switch between different codecs, change between transport protocols used (TCP or UDP) or even adjust the level of FEC applied. In the paper [63] such a method for adaptive QoS control for VoIP is suggested and verified by means of an experimental implementation. In their work they show that in some cases the suggested method gives a measureable quality improvement, while in other cases the effect is neutral. In order to use this method, the involved clients must support a range of different codecs and also additional system level software for control.

2.5.3 Rate Adaptive Video

The development of concepts and supporting protocols for adaptive video streaming across the Internet started many years ago within the IETF. This work resulted in a whole family of specifications around the three main ones which are the Real-time Transport Protocol (RTP) [64], Real-time Streaming Protocol (RTSP) [65] and Real-time Transport Control Protocol (RTCP) [66].

The RTP protocol provides end-to-end network transport functions for real-time multimedia content, but has no means of providing interactivity to the clients involved. In order to achieve such interactivity (e.g. start/stop/pause) – the RTSP protocol has to be used. Hence, RTSP can be considered an extension of the RTP protocol. Both RTP and RTSP can use RTCP for probing the quality of active sessions, which then can be used to perform adaptations of the quality or transmission rate on the session level.

Video streaming solutions based on RTSP have been around for some time. However, they have not become part of today's more popular multimedia services on the Internet. The reasons for this are related to interoperability, requirement for supporting new protocols and also to some extent complexity. Instead, the use of HTTP for transport of video has become the dominant solution for delivering this type of services to the mass market. In particular, the recently standardized DASH [67] concept is today preferred by many video content providers on the Internet (e.g. YouTube [68], Netflix [69] et.al.).

Before the first version of the DASH specification was published by the Moving Pictures Expert Group (MPEG) in 2012, there was a strong drive in the market in terms of starting to promote and also use proprietary solutions in this domain. The main commercial providers of content independent solutions were Microsoft (SilverLight Smooth Streaming), Apple (HTTP Live Streaming) and Adobe (HTTP Dynamic Streaming). The common parts of their solutions are that they all use HTTP for transport and also the ability to switch between quality levels during sessions. However, they use different manifest and segment formats which prevents interoperability. The implementation of the DASH specification on the interface between client and server will eventually resolve these issues. Some of the functional and performance related differences between the pre-standard solutions are shown in [70]. Significant differences are found in terms of how often the DASH clients are allowed to request a change in the video streaming quality level. The Apple solution tries to maintain a stable quality, at the expense of average quality level – by not allowing very frequent quality changes (typically once per 10-15 sec). The Adobe solution follows a very different strategy, allowing quality changes almost continuously. This would normally give a higher average quality level, but at the expense of potential quality fluctuations. The Smooth Streaming solution is somewhere between Apple and Adobe in terms of how often quality may change (typically once per 2 sec).

One aspect of DASH solutions which is not likely to be subject for standardization is the controlling algorithm inside each client which decides whether the quality level should be changed. None of the available solutions from Microsoft, Apple or Adobe disclose such details as it represents parts of their intellectual property and potentially competitive advantage. Intuitively, one would think that the client in some way, either directly or indirectly makes some kind of estimation of available bandwidth between the client and server. However, as the accuracy and real-time capabilities of such methods can be questioned [71] [72] it may be that a somewhat more basic approach is used by adaptive video streaming clients. Independent of what is being used in the commercial solutions the topic has been given quite a lot of attention in the research community over the last years.

For a client to estimate available bandwidth and use this as input to its quality selection it must either perform active or passive measurements, or be able to deduce similar information from the video traffic itself. In [73] a rate adaption algorithm is proposed which is based on the smoothed average of segment fetch durations. It another work [74] the potential use of feedback control theory for this purpose is discussed. An experimental comparison of this approach and live services provided from the Akamai CDN shows that the suggested method is better able to handle bandwidth transients with durations of less than 30s. However, as it is not stated explicitly which streaming solution was used on the Akamai side – their findings may not have general applicability.

In this thesis work a new method for estimating available bandwidth is presented, which utilizes the periodic characteristics of traffic originating from DASH services (cf. paper: P8, P9 and P10). The method can be used by DASH based services themselves, in the case where they compete against other DASH sessions for a shared resource – or by other Internet applications / services which have the ability to adapt their capacity requirements.

2.6 Transport Protocols

It was the common understanding in the Internet community for many years that UDP as transport protocol was the best choice for streaming audio or video. The TCP protocol was not considered as suitable due to the use of retransmissions, which could lead to unacceptable playout latency – and also the congestion avoidance mechanism, which could cause abrupt rate variations. However, even UDP had issues – among which the quite frequent problem of firewall traversal was one.

The choice of TCP or UDP for streaming multimedia has been subject for a lot of research over the years, from different perspectives. In an early work [75] the case of streaming multimedia with TCP was analysed in the context of adaptive video. The conclusion in this work was that TCP was a viable and effective choice for the specific context at hand. Although the adaptive video solution used in their study does not match the DASH concept of today, there are enough similarities to make the research quite relevant even today. In a more recent study [76] more general findings are presented based on simulations stating that multimedia streaming based on TCP gives satisfactory performance in terms of startup delay, video playback rate and on a lower level – packet loss rate.

The TCP versus UDP discussion has also been impacted by external factors. On the client side, there has been a significant evolution in terms of both CPU and buffering capabilities. When this was combined with increasing access capacity towards the

Internet, a more active use of receiver side streaming buffers became viable. The use of such buffers of increased size solved a lot of the problems with retransmissions and rate changes when using TCP for streaming. Thus, more powerful clients and the broadband evolution made TCP more attractive.

2.6.1 The evolution of TCP

Looking closer into the TCP protocol it is important to note that there is no such thing anymore as "the TCP protocol". The starting point for TCP was back in 1988 when the foundation [54] for the TCP Reno protocol [77] was developed. Following this, several categories of TCP protocols have emerged. The protocols are categorized based on how the congestion control mechanism is implemented [78]. The TCP Reno protocol had a congestion control which was purely based on packet loss. New flavours of TCP emerged which use delay, a mixture of loss and delay or even explicit congestion notification feedback [79] [80] as basis for congestion control [78]. All these new TCP protocols have dramatically increased the complexity in the discussion on TCP versus UDP for streaming audio or video. Further on, it has become a concern that the new TCP protocols are too aggressive [81] and have unfortunate effect on other traffic.

Another interesting aspect of TCP versions which is important to notice is that the differences between them all reside on the server side. The congestion control mechanisms decide the per-session congestion window (CWND) on the server side, and this parameter is not visible to the client. The client only knows his receive window (RWND). This makes it possible for any server on the Internet to use whatever TCP protocol without being concerned whether the client side supports the specific version.

For server type platforms the major operating systems used today are Linux, Solaris and Microsoft. These all support a wide range of TCP versions, which are configurable on the OS level. However, unless multiple protocol stacks are configured it is normally the same TCP version which applies to all TCP based communication towards a specific server. This is something which must be kept in mind if one considers using protocol versions which are optimized for one specific service, on a multi-service server platform. In many Linux distributions (e.g. Ubuntu) the TCP CUBIC [82] is the default version, while on Solaris the TCP Fusion [83] is commonly seen as the default version. For Microsoft Servers the Compound TCP (CTCP) [84] is normally used.

2.6.2 TCP CUBIC

The TCP CUBIC protocol was developed for high-speed networks, and uses only packet loss as input to its congestion control. Although it was designed for networks with high bandwidth and high latency (commonly known as Long Fat Networks), it has been shown that it also performs well in the case of low bandwidth or low latency in terms of TCP-friendliness [78].

The performance of TCP CUBIC when used for streaming video was studied in [85] and compared with other selected TCP versions. The performance aspects used in the comparison were discarded pictures (as reported by the video player) and TCP retransmissions on the network layer. In this work the findings for TCP CUBIC was positive in most network scenarios studied, except for a wireless link environment with random packet loss events. In this case, the delay based TCP versions used in the study performed better.

2.6.3 Hybrid TCP

The Compound TCP and TCP FUSION protocols are hybrid versions which uses both delay and packet loss as basis for congestion control. The objective of this hybrid approach was to combine the best parts from two worlds, i.e. both the delay based and loss based TCP versions. The desired characteristics were bandwidth scalability and round-trip times (RTT) fairness, while maintaining TCP-fairness. The concept of RTT fairness refers to the goal of equal bandwidth usage among competing flows with different RTT, across the same congestion points.

As stated in the findings of [85] the Compound TCP protocol in some cases performs better when used for video streaming than pure loss based protocols such as TCP CUBIC. The good performance of Compound TCP in scenarios with random packet loss is also supported by the findings in [86]. However, these findings do not hold across all network and usage scenarios, and therefore it is not possible to make a general statement with regard to which TCP version is best suited for video streaming.

2.6.4 TCP in CDN platforms

As information about TCP version used remains hidden inside the server side OS, it creates an opportunity for commercial actors to differentiate themselves. Not all TCP versions are freely available for use as they are protected by patents. An example of such a protocol is FastTCP [87], which is protected by several patents owned by Akamai – one of the leading CDN operators in the world. The FastTCP protocol is a delay based TCP version, which uses a mathematical optimization problem as basis for continuously computing the path delay and thereby the appropriate congestion window size. Whether the protocol is in active use of not is not public available information.

This thesis work does not address any specific TCP version used by DASH. In the first paper (cf. paper: P1) a method for controlling DASH quality levels by delaying TCP ACK packets were studied, and the testing of this was done using a live video service provided from the Akamai CDN.

3. Research Design

The research has been conducted as part of the R2D2 project, which was a Norwegian research project with participants from industry, research institutes and universities. The scope of this project was to develop techniques and tools for modelling, analysis and evaluation of users and usage patterns, in order to enable the service providers to deliver according to specific QoS and QoE requirements. The project had specific work packages for traffic analysis by means of measurements, and demonstration of new functionality in a testbed environment. This served as direction for the research both in terms of goal and methodology.

The organisation of the R2D2 project allowed for the thesis research to be planned and executed in autonomous way, but at the same time providing support when needed. The active involvement and interest from the commercial content and network providers was quite valuable. The adaptive video streaming solution from Microsoft [88] was used in the testbed because this was the solution the commercial project partners were using and had most interest in.

3.1 Research Goal

The goal of the research documented in this thesis was to propose, and if possible - validate models and methods for optimization of quality aspects relevant for video services delivered Over-The-Top. The scope of the optimization effort covered service quality on the individual user basis / per session basis and for a typical home network with a limited amount of users present and a shared Internet access. An important aspect of models and methods to be investigated was that they should not rely on any functionality in the network operator domain. Thus, the main focus was the service endpoints, i.e. client / server – whereas the latter could be represented by a CDN.

It was the intention to include quality metrics from both the QoS and QoE domains as basis for evaluating the suggested methods. If possible, relationship between QoS and QoE metrics should be suggested and potentially verified. The starting point for investigating new methods should be on the network and service level, using well known QoS metrics which are possible to observe and measure. Thereafter, it was a goal to establish hypothesises regarding how these metrics could influence subjective QoE metrics.

3.2 Research Model

The structure of my research was inspired by a model used in the Multi Service Access Everywhere (MUSE) Project "Advanced features for MM enabled access platform" [27], which was part of the 6th framework programme funded by the European Commission. In their model for a multimedia enabled access network supporting advanced features the different functionality were grouped according to membership in the Knowledge Plane (KP), the Monitor Plane (MP) or the Action Plane (AP).

MP: Measurements of different aspects of a service or an element.
KP: Composition and reasoning of collected and exchanged information
AP: The execution of operations aimed at improving QoS and/or QoE

In order for this model to fit OTT service delivery, I extended it to include both the client and service provider elements (cf. Figure 2). Although not normally used for OTT, the presence of KP/MP/AP elements located in the network operator are shown – for the purpose of at least illustrating that interested operators could have an opportunity to take a more active part in OTT value chains.

Based on this model I have identified research topics which contribute to answer the defined research questions. Rather than putting all the effort into only one part of this model, the intention was to cover it end-to-end.

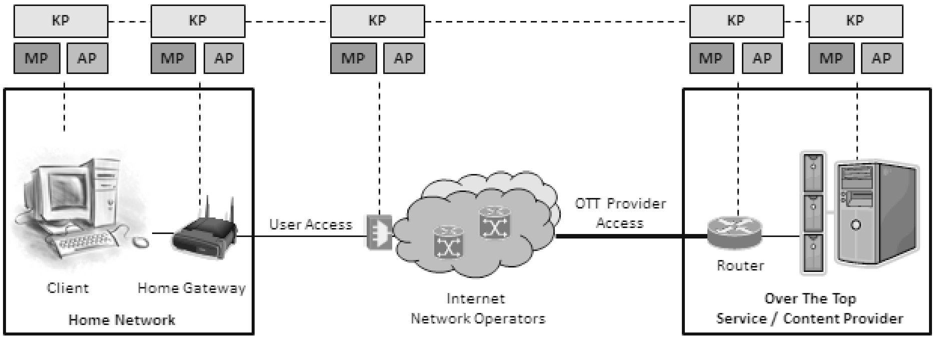


Figure 2: Research model for OTT service delivery

In a home networking environment the amount of different devices used to access services are growing quite rapidly. A key component in these networks is the home gateway device which provides the Internet connectivity and also provides other services to end user devices (e.g. DHCP, DNS and firewall). Although some network operators include the home gateway as part of their infrastructure, it has been assumed to reside under the control of the home network in the research model. This scenario is in line with the view of organisations such as the Digital Living Network Alliance (DLNA) and Universal Plug and Play (UPnp) Forum which are working to achieve seamless connectivity, simplified configuration and improved end user experience when using services in a in a home network environment.

On the server side in the research model the OTT service provider is located. This could in some cases be a dedicated infrastructure for one specific service provider, but in the case of video streaming on the Internet is has become quite common to host this type of services in CDN networks. In addition to the general benefits of a CDN for providing bandwidth demanding services, it also makes it possible to consider new methods for optimizing services delivered to the same home network environment. Reason being that the CDN can easily identify competing video sessions being delivered to the same home network based on the source IP addresses.

3.3 Research Method

The methods used to analyse the different topics throughout the research period has been chosen based on the nature of each specific topic. The main research methods commonly seen in this domain are based on analytical modelling, experiments using measurements or simulations. Choosing between these methods can be done based on different criteria, among which the life-cycle stage is one of the more important ones. In other words, for new concepts or algorithms to be studied, only analytical modelling or simulation is possible - while for existing systems (including prototypes) the alternative of experiments using measurements is also possible. Some of the other key criteria to be considered are the time required for doing the analysis, tools available, required accuracy and cost [89]. Whenever possible, combinations of methods are quite useful and strengthen the findings by cross-validating the results under different assumptions.

The papers included in this thesis are mainly based on simulations and experiments, but with some analytical supplements whenever feasible. The reason for this choice was the desire to demonstrate new methods in a R2D2 project testbed, and also the availability of the systems subject for optimization.

3.3.1 Simulations

The planning of a simulation study should contain many important aspects [90]. The starting point is a thorough understanding of the system to be modelled in the simulator, followed by the selection of input variables and parameters, and also the output metrics. It is important to remember that the more input variables that are used, the more complex the statistical analysis will become. Thus, a good advice is to eliminate as many input variables as possible during the development of the simulator, leaving in only those which have the most impact on the output metrics. Concerning the output metrics, one has to choose those which best represents the aspects (e.g. performance) of interest in the study. In addition to the selected output metrics, which will be subject for monitoring and potentially statistically analysis, maintaining some additional output metrics just for the purpose of verifying model and simulation validity is recommended.

There are different types of simulation approaches available which could be applied. The main ones which could be considered relevant for the research questions in this thesis are Monte Carlo Simulation, Trace-Driven Simulation and Discrete-Event Simulation [89]. The Monte Carlo approach applies to static simulations without a time axis. Thus it would apply to probabilistic phenomenon which does not change behavior over time. The trace-driven approach is the scenarios where a real system trace is used as input to the simulator. The trace consists of a time-ordered list of system events, and does in this regard not contain any probabilistic aspects – except for the case where the system itself generating the trace is driven by probabilistic aspects. An important requirement for this approach to be applicable is that the input trace used is independent for the system under study. The potential credibility of the results from a trace-driven simulation is quite often stated as a reason for choosing this, but at the same time concerns about representativeness of the results pulls in other directions. The reason for this is that a specific trace only represents the system behaviour in a specific scenario described by aspects such as workload, traffic mix and protocols used. The discrete-event approach applies in cases where the system being studied can be described in terms of a discrete-state model, and the time axis may use either discrete or continuous values. The inputs to a discrete-event simulation during run-time are in most cases probabilistic, but for the purpose of model validation it also allows for deterministic input.

In simulation studies of adaptive video streaming where selected quality aspects are used as output metrics, the time aspect is indeed relevant. Thus, the Monte Carlo type of simulation was not considered as appropriate. However, both a trace-driven and a discrete-event approach could have been applied. For the purpose of seeking results of wider representativeness the discrete-event approach was chosen. The discrete-state models on a per video session level followed directly from the quality levels available for the specific videos.

Having chosen the type of simulation approach, the next step is to choose the proper programming language or tool. The choices are either a simulation language, a general purpose programming language, extensions to general purpose languages or some kind of simulation package. There is no absolute right or wrong choice in this field, but it is important to make a choice which has the best chance of giving useful and valid results. The high-level simulation packages are to some extent appealing for certain purposes as it may not require extensive programming, but at the same time they quite often operate at a high abstraction level with not enough control of details in the model. In cases where standard communication protocols are important to includes as part of the simulation, the simulation packages are recommended as these quite often have this type of functionality available as ready to use functionality. In cases where one wishes to really have full control of all aspects, building the simulator from scratch using a general purpose programming language such as C/C++ is recommended. However, in most cases there are strong reasons for using a simulation language for the task at hand, as this provides the programmer with a range of commonly used and acknowledged modules for things like random number generation and probability distributions.

The simulation languages considered for use in the simulation studies conducted as part of this thesis was NS-2 [91], NS-3 [92] or Simula/Demos [93]. The most common ones of these are the NS versions, as they provide a wide range of modules and also allows for low level customization through scripting and custom programming. However, for NS-2 there are general concerns about the code quality due to the way it is being documented and maintained. In the case of NS-3 which is a more recent simulation language, it still does not contain all the functionality which was available in NS-2 as it has been built from scratch for the purpose of ensuring higher code quality (among other things). The Simula/Demos simulation language is actually a combination of the general purpose programming language Simula and the Demos context class which a set of commonly used functions applicable for discrete event simulation. Compared to NS-2 or NS-3 the Simula/Demos language is far from as feature rich and does not e.g. have modules available for integration of communication protocols in a simulator. Thus, if such is required as part of the simulator the best choice would be either NS-2 or NS-3. However, if this is not the case and low level control of the discrete event simulation is required, the Simula/Demos language is quite attractive. The language is well documented and easy to use, at the same time as it allows for post-processing of results by other more advanced tools. Based on this, the Simula/Demos language was chosen for the thesis work which required simulations.

3.3.2 Experiments

In order to perform an experimental analysis of a certain topic one can either do it in a live network and service scenario, or in a controlled environment but still with real services. The first approach is the one which could be stated as most realistic as it would have real background traffic, something which is challenging to generate in the more closed and controlled environment. However, in most cases it would be best to conduct the initial experiments in the controlled environment as the effect of algorithms or methods being investigated potentially would be easier to detect in this case. Another aspect which also speaks in favour of using a controlled environment for experiments is that reproducing results are easier. The experimental analysis done as a part of this thesis work was done in a controlled environment.

3.3.3 Network Components

In an experimental network which are to be used for different purposes there is a need for both commercial off-the-shelf (COTS) and more programmable components. The main contribution of the COTS components would be to provide access to key functionality used in live networks such as e.g. QoS profiles. This contributes to make the experimental network as realistic as possible, and also reduces the amount of work required to make the network operational. Another benefit of including COTS products in the experimental network, if properly chosen, is that they can provide useful capabilities such as port mirroring / network taps. In order to measure transit traffic in a non-intrusive way such a capability is the preferred option. In the experimental network used in this thesis research, the COTS components used were from Cisco Systems.

The use of programmable components in the experimental network allows for the implementation of new mechanisms alongside with standard functionality. For this purpose the Click software router [94] is a good solution as it allows you to build your own complete router with full control of each IP packet being processed. This solution can be run on almost any standard PC platform with multiple interfaces and various types of Linux OS. As part of the SW distribution there is a rich library of functions which can be used, and it can also be extended by programming in C++. The Click software router can be run in either User Level or Kernel Level. In the latter mode, it takes over the kernel routing and interface packet handling completely – and is then capable of doing the role of a regular router but with selected additional or customized capabilities. Another interesting capability of Click, which contributed to the choice of using it in the testbed as home gateway router, is the capability to communicate with external entities during run-time. Thus, allowing for external control of the experimental router.

3.3.4 Service Endpoints

In the majority of experiments performed, the adaptive video streaming solution from Microsoft was used. The Microsoft Smooth Streaming solution is not yet MPEG-DASH compliant, thus it requires the client also to be of the same type i.e. having a browser which allows the Silverlight plug-in to be installed. Initially, this meant that the client operating system had to be Microsoft with Internet Explorer as browser. This changed to some extent in line with the Microsoft intentions for Silverlight to become a cross-browser and cross-platform plugin for web based media and interactive applications. Silverlight is now supported also in Mac OS, and other browsers such as Firefox, Safari and Chrome. However, for adaptive streaming to Mac OS based devices and iOS based devices (e.g. iPAD's), the adaptive streaming solution from Apple themselves is more common. The Apple solution for adaptive video streaming is called Apple HTTP Live Streaming (HLS) [95] and this was also not yet MPEG-DASH compliant during the research period for this thesis.

The server side providing the streaming service was a Windows Server 2008 in the controlled environment (tested), while it in real life normally would be a part of a CDN. The Windows 2008 server allowed for simultaneous delivery of both Smooth Streaming and Apple HLS based content. In order for a single video file to be available from the server in both formats on the server it requires it to be encoded separately for each streaming solution and with the (still) proprietary manifest files for each solution.

In order to repeat experiments many times, using e.g. different network or service configurations it is very convenient to have remote control of all devices used in the testbed. For most devices such as routers, switches and servers this is rather straightforward as they support some kind of remote login providing full system access. However, for clients such as PC's with Windows OS this is more challenging. The standard way of taking control of a PC using remote desktop is not something which can easily be incorporated in a script oriented controller node. In order to facilitate system level access to a Windows PC the Cygwin [96] SSH capabilities was used. Once installed on a PC and allowed through the system firewall, the controller node in a testbed can SSH into the involved PC's and start/stop applications such as web browser using specific url's.

In cases where testbed components are connected to the Internet it is important to control all software updates manually, both on OS and application level. If this is not done it is likely that both functional and performance related aspect will change during the period of experimentation. If so happens, measurement errors can be introduced which reduced the validity of the results. Another reason for making sure that such updates are not being done during an experiment is the potential additional load it represents on both access link and system CPU. Such aspects are also a potential source of measurement errors. In general, any change in a testbed either on the physical or logical level during an experiment should be avoided, unless if it is a planned change representing a variable of the topic being investigated.

3.3.5 Traffic Generation and Analysis

When doing experimental research on networks and services there is potentially a need for both traffic generators and analysers. A traffic generator can be used for different purposes such as generating background traffic, generate special traffic of interest or even as a network probe of some kind. The traffic analysers serve several important purposes such as verifying communication patterns, traffic load, detection of interesting events or even as collectors of detailed packet traces for the purpose of in depth post-processing and analysis.

For both traffic generators and analysers there are a lot of commercial products available which has been developed for these particular purposes. In many cases they operate on both custom HW and SW, and deliver a performance according to set of specifications. Using such tools is quite appealing, as it removes a lot of uncertainties regarding the tools being used in the research. However, as such commercial products are quite expensive they are not commonly seen in the academic research environment. Therefore, the use of open-source SW solutions for both traffic generation and analysis on standard HW platforms is of great interest. This reduces the cost, but introduces a concern about general performance, and maybe accuracy in particular. The challenge is to make sure that a standard multi-purpose HW platform with a standard OS installed is configured in an optimal way for a single purpose i.e. to act as either a traffic generator or analyser. This challenge must be addressed for the specific SW solution to be used. As stated in [97] there are several sources of interference for the different tools available in this domain, which are not necessarily easy to control. Understanding them is essential in order to secure quality in results.

When using a standard OS as basis for a traffic analyser in a scenario where timing aspects are of interest, it is very important to understand how the OS relates to the network interfaces. If not special NICs are used which can apply timestamps at the lowest level, the traffic analyser SW will add the timestamp to packets received when delivered from the NIC to the OS kernel. This would bring our attention to how often packets are delivered to the kernel. For multi-purpose platforms and standard OS versions, it is common to limit how often the NICs are allowed to interrupt the kernel for the purpose of delivering packets for processing. The term used for this capability is interrupt coalescence and is discussed in detail in [98]. In order for a traffic analyser to add as accurate timestamps as possible, it is desirable to remove this limitation, thus allowing the NIC to interrupt the kernel for each incoming packet. Thus, reducing time spent in NIC buffer for each packet and then also significantly improving the accuracy for timing aspects. Tuning of interrupt coalescence was an important requirement for the work reported in paper P10 included in this thesis.

The Rude/Crude package [99] is a SW based command line tool for traffic generation and collection which can be installed on a Linux platform. Although the package is quite a few years old, it is still widely used and has been shown in some cases to provide a high degree of accuracy [100]. The traffic generator (Rude) only supports UDP, which to some extent limits its capability to generate realistic background traffic. However, for the purpose of being used as a network probing tool UDP is sufficient. The support of application level timestamps both on the sender side (Rude) and receiver side (Crude) makes the tool very attractive to use when one-way-delay for packets through a network is of interest.

A more general purpose tool for understanding and measuring aspects of IP traffic is Wireshark, which was originally named Ethereal. There are both GUI and CLI based versions of this. The CLI based versions (tcpdump, tshark) are more convenient to use in automated measurements driven by a controller node. With regards to filtering capabilities, the display filters used by the GUI versions are more powerful than the basic capture filter. This leads to that a certain amount of post processing in most cases are required in order to extract the specific information of interest. For this purpose the capabilities of Linux OS are quite powerful. The combination of command based utilities, shell scripting and awk was used to automate also the post processing of measurement results. For the purpose of generating previews of results, the Gnuplot utility was used to generate graphical summaries on a continuous basis.

For more in-depth analysis of measurement data captured in the testbed during an experiment, the use of tools such as Mathematica was required. Especially for time series analysis on large datasets this tool was of great help. Mathematica also served as an important tool in the more analytical parts of the research, e.g. when solving optimization problems.

4. Results

For each research question a number of related sub-topics was identified and made subject for analysis. This analysis resulted in published papers, with findings which together constitutes the thesis answer to a specific research question. In all cases, the end result of answering the research questions leads to research contributions.

4.1 Answers to research questions

4.1.1 Research Question 1

In a home network how to (autonomously) control the performance of DASH based services ?

The thesis answer to this question is composed by the findings in papers P1, P2 and P3 as illustrated in Figure 3. In addition, the findings in paper P11 is considered related but does not directly contribute to the RQ1 answer. The RQ1 answer leads to contribution C1.

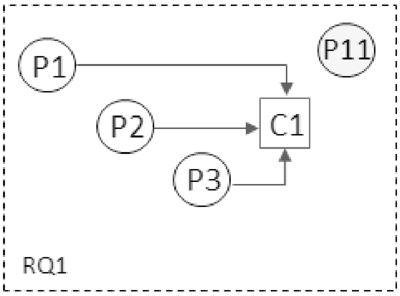


Figure 3 Contributions to RQ1:
In a home network how to (autonomously) control the performance of DASH based services.

In the P1 paper *Monitoring and Control of QoE in Media Streams using the Click Software Router* [101] a mechanism for controlling the rate of TCP flows is analyzed by means of experiments in a testbed environment. The mechanism investigated is called PostACK and is based on selectively delaying TCP ACK packets for a specific flow of interest, for the purpose of triggering TCP congestion avoidance mechanisms and thereby making the TCP sender to adjust its sending rate. The mechanism is implemented in an experimental home gateway router using the Click modular router [94] software. The service of interest to see if could be controlled by this mechanism was adaptive video streaming over HTTP. The results show that for the specific scenario investigated, it is possible for a home gateway to influence which quality level is requested by a specific client located on the home network.

The significance of the PostACK mechanism investigated in P1 with regards to RQ1 is to show that the home gateway node can participate in controlling the performance (bitrate levels) of individual DASH sessions, in a potentially less resource demanding approach than traditional rate limiting functions.

In the P2 paper *Towards Knowledge-driven QoE Optimization in Home Gateways* [102] the potential performance (average bitrate) gain of making a home gateway aware of the different quality levels available for DASH sessions, together with service preferences – is investigated. As indicator of performance gain, the average achieved bitrate for the preferred service is used during times of congestion. Three different schemes of implementing a bandwidth broker [103] function in the home gateway are analyzed by means of simulation. The scheme which fully utilizes the knowledge about available quality levels for the competing DASH sessions is called STEPDOWN. This scheme resolves congestion and allows the preferred service to improve its performance by forcing other non-preferred DASH sessions to go down in bitrate to the next level below the current one. The basic results show that the method provides improved performance over a wide range of congestion levels on the access link.

The motivation for locating a bandwidth broker function in the home gateway is based on the natural role of this component as the controlling entity for a home network. Thus, it contributes to the research question RQ1 by demonstrating how information about competing sessions can be used to optimize video session performance (bitrate level and average bitrate).

In the P3 paper *A Monitor Plane Component for Adaptive Video Streaming* [104] different alternatives for making a DASH client share information with external entities such as the home gateway about which quality levels are available for a current session are discussed. Further on, a prototype implementation [105] for the Microsoft Smooth Streaming solution is described. This prototype is based on extending the Silverlight code running in the customer browser with functionality to send reports to specific destinations on regular basis, or whenever events such a quality change requests occur.

The input from the P3 paper with regards to RQ1 is to show how the home gateway node can obtain information about current DASH sessions without having to implement demanding functions like deep packet inspection or similar for transit traffic.

In the P11 paper *Investigating Quality of Experience in the context of adaptive video streaming: findings from an experimental user study* [106] one of the things which are studied is the impact on end user QoE from different bitrate profiles. The relevant findings with regard to RQ1 is that end users clearly appreciate streams of stable quality rather than streams with a higher average quality level but with sudden and big changes.

In summary, the answer to RQ1 based on the aggregate findings of the referred papers is that autonomous control of DASH services delivered to a home network is possible using the PostACK method. The underlying assumption is that information (quality levels available and current level) can be made available about each session to the home gateway, and a method for this is described in P3. The home gateway node is identified as the candidate controller element for a home network, and examples of how chosen performance aspects can be optimized are shown by means of experiments and simulation. This represents contribution C1: A method for controlling the quality levels of DASH in the home gateway, which will be subject for further discussion in Section 5

4.1.2 Research Question 2

Is it possible to achieve increased fairness, while retaining stability for competing DASH sessions without involving network components between the client and the server ?

The thesis answer to this question is composed by the findings in papers P4, P5, P6 and P7 as illustrated in Figure 3. In addition, the findings in paper P11 is considered related but does not directly contribute to the RQ2 answer. The RQ2 answer leads to contributions C2 and C3.

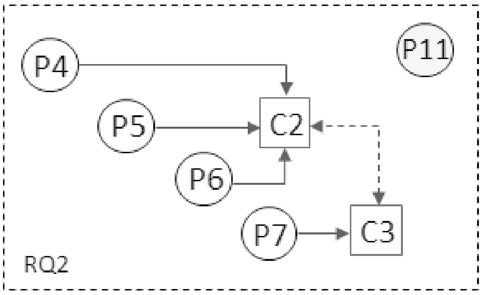


**Figure 4 Contributions to RQ2:
How to provide fairness and stability for competing DASH sessions using service endpoint functionality.**

In the P4 paper *Improving Perceived Fairness and QoE for Adaptive Video Streams* a method for achieving improved fairness among competing DASH session is studied. The method is based on changing the default fixed time interval between each potential rate adjustment, over to a time interval decided by a stochastic process. Fairness is studied from both network and end user perspective. From the network side, the fairness metric used is proportional fairness, while from the end user side a hypothesis on perceived fairness being related to the frequency of rate changes is presented and used [107]. The simulation study of these metrics when applying the suggested changes to the time intervals between rate changes shows that the potential improvements are significant. The results from the simulation are further supported by analytical considerations which explain in more detail why the method has effect.

The significance of the findings in P4 with regards to RQ2 is to show that it is possible to improve fairness and maintain stability (in this paper called perceived fairness) from a theoretical perspective. The findings justified more work to be spent on the specific method, and as the next step an experimental study was chosen.

In the P5 paper *Improving Fairness for Adaptive HTTP Video Streaming* the same method as suggested in P4 is in this paper studied by means of experiments. However, some slight modifications to the method are made in order to make it possible to implement. The approach of having a stochastic interval between each potential rate change is not practical to implement. The reason for this is that each valid interval corresponds to a specific encoding of a video. Thus, allowing the amount of available intervals to be very high would require a lot of encoding effort and storage space per video. The modification required of the method in order to implement it in the testbed was therefore to make available a total of 10 different interval values, and limit the stochastic selection process to operate on this set. Further on, a slight variation of the method to apply a per session unique interval value is also included in the study. In the experiments a typical home network environment is used with 5 competing DASH sessions. The results show that both the random interval and unique interval approach give improvement in intra-session fairness. The experiments also show that the frequency of rate changes does not increase due to the changes in intervals used.

The significance of the findings in P5 with regards to RQ2 is that it brings the method originally suggested in P4 closer to a practical implementation and also supports the initial findings.

The P6 invited journal paper *Improving Fairness in QoS and QoE domains for Adaptive Video Streaming* is a combined study where the simulation and analytical model in P4 is harmonized with the experimental scenario from P5. The harmonization is done both in terms of the user scenario (number of users, access capacity) and interval selection method. The results from the simulation part deviate in absolute values from the original simulation in P4, however the findings still remain – the method improves proportional fairness while maintaining perceived fairness. These findings are supported by both the simulations and the experiments.

The significance of the findings in P6 with regards to RQ2 is that it presents a more complete analysis of the original method as suggested in P4, with modifications making it possible to implement in real life e.g. in a CDN network delivering many DASH session to the same home network.

In the P7 paper *Group Based Traffic Shaping for Adaptive HTTP Video Streaming* the traffic shaping effect of the method suggested in P5 is investigated. The motivation for this work was two-fold. First of all, it was of interest to better understand and document the method's effect on the aggregated video traffic. My hypothesis was that the improvements in fairness among competing sessions could be related to an underlying shaping effect. Second, investigating if the suggested method gives a shaping effect of video traffic aggregates also represented an interesting topic to investigate on its own.

The significance of the findings in P7 with regards to RQ2 is that it provides additional insight with regards to the effects of the method applied in P4, P5 and P6. The shaping effect is of interest by itself but also as a potential underlying reason for why the fairness improvements are seen.

In the P11 paper *Investigating Quality of Experience in the context of adaptive video streaming: findings from an experimental user study* [106] one of the things which are studied is the impact on social setting on end user QoE. It is investigated whether users are more sensitive with regards to quality changes when they are given a notion of access capacity sharing. The idea of this was partly to see if the presented hypothesis regarding perceived fairness as suggested in P4 was valid. The users involved in the experiments were exposed to the same video bitrate profiles (stable high, changing, stable low) in two different usage scenarios. In the first scenario each user was sitting alone with a dedicated own access to the Internet (in reality the video server in the testbed). In the other scenario the users were moved together into a group and informed that they were now sharing the access to the Internet. The quality ratings given by the end users for the two usage scenarios did not give firm answers to whether the social setting has an impact on end user QoE. Thus, the hypothesis concerning rate change frequency as a perceived fairness metric could not be rejected or accepted based on this work.

In summary, the answer to RQ2 based on the aggregate findings of the referred papers is that improved fairness among competing DASH sessions, while retaining stability is possible to achieve by the suggested method. The method is also shown as possible to implement in an OTT scenario since the only change required can be done on the CDN side. Although not discussed directly in the papers, in the case of DASH services being delivered from a CDN, the most practical way of implementing the method is to apply different rate change intervals to different videos, rather than having many intervals for each video. Assuming competing session are not the same video, the effect would then be the same. This approach would give minimum encoding effort and storage requirements. This represents contributions C2: A method for improving fairness among competing DASH sessions and C3: A method for shaping traffic aggregates on access links with DASH components, which will be subject for further discussion in Section 5.

4.1.3 Research Question 3

What are the control challenges for a DASH session with regard to the choice of quality level and how can it potentially be improved ?

The thesis answer to this question is composed by the findings in papers P8, P9 and P10 as illustrated in Figure 5. The RQ3 answer leads to contribution C4.

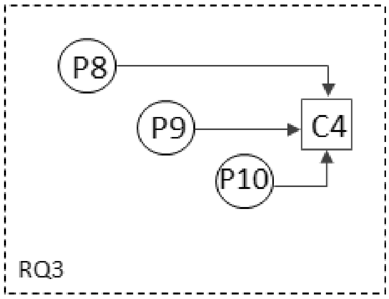


Figure 5 Contributions to RQ3:
How to choose appropriate quality levels for a DASH session.

A DASH client chooses which quality levels to receive from the video server side on a regular basis based on certain criteria's and observed metrics. Exactly which metrics are being used is not disclosed by vendors but it is likely that aspects such as buffer filling, cpu load and estimated available bandwidth are used by some. The answer to RQ3 in this thesis is focused on the latter, i.e. how to estimate available bandwidth on access links as accurate as possible. Thus, the answer implies that using an available bandwidth estimation scheme inside each DASH client can be a contributing control component on the session level.

In the P8 paper *Detecting Period and Burst Durations in Video Streaming by means of Active Probing* the foundation for a new approach to estimate available bandwidth is presented. The idea is that when video has become the dominating service component on the Internet, it could be beneficial for available bandwidth estimations to utilize some key characteristics of such traffic. The delivery of video towards a client is based on sequences of segment fetch requests sent from the client. These sequences generate periodic traffic patterns containing burst components. In the P8 paper a method for detecting such periodic behavior is presented which is based on active probing and serial correlation of probe packet inter-arrival times. The output of the serial correlation not only indicates the period (if present) in the cross-traffic, but also the duration of the burst component. The findings presented in the paper are quite positive in the sense that by using only very moderate amounts of probe traffic it is possible to detect both periodic behavior and burst duration. The experimental analysis was done for both controlled lab services and a live Netflix service. Although the Netflix service operates in a special way using multiple TCP session, the probing method still gave valid output.

The significance of the findings in P8 with regards to RQ3 is that it forms the basis for a new active probing method using knowledge of cross-traffic period and burst duration in order to align probe traffic. The purpose of aligning the probe traffic to a cross-traffic profile is to get as much information out from each probe packet sent into the network as possible.

In the P9 paper *A Measurement Study of Active Probing on Access Links* the accuracy of different active probing methods are investigated. The methods investigated are packet pair probing and single packet probing. The first approach uses measured increase in packet spacing as indicator of cross traffic, while the latter uses increase in one-way-delay for probe packets as indicator. Each of these methods have their own strengths and weakness, both in terms of timing requirements and amount of post-processing required. The findings illustrate the strengths of a probing method based on one-way-delay measurements, under the condition that the required timing accuracy is achieved and delay characteristics is available for the involved network path. The challenge of providing an accurate timing scheme (NTP versus PTP) and delay characteristics are not trivial. However, the results when using PTP as time synchronization protocol are quite promising and speak in favor of the single packet probing method.

The relevance of the findings in P9 with regards to RQ3 is that it addresses active probing accuracy in general. By doing this it provides input to both the process of probing for the purpose of detecting period and burst duration as described in P8, and also the stratified probing method as a whole which is covered in P10.

In the P10 paper *Estimating Available Bandwidth on Access Links by Means of Stratified Probing* a new approach for estimation of available bandwidth on access links using stratified probing is presented. The challenge of performing such estimations in this network part is related to the bursty nature of cross-traffic and the related uncertainty regarding appropriate time period for producing sample estimates. The method suggested in this paper is based on a four-phase approach. In the first phase a traffic profile for the cross-traffic is established, with focus on detecting periodic behaviour and duration of respectively burst and idle sub-periods. This topic is described in detail in paper P8. In the second and third phase, the active probing is split into strata and synchronized according to the burst/idle sub periods. In the final phase, the actual probing and estimation of available bandwidth takes place. The method is analysed by means of experiments in a controlled lab environment. The measurements results are considered quite promising both in terms of accuracy and the low degree of intrusiveness facilitated by the stratified approach.

The relevance of the findings in P10 with regards to RQ3 is that it describes and demonstrates the capabilities of a stratified probing approach. It uses findings from paper P8, and can also be further enhanced by using the findings in paper P9 i.e. changing the probing approach from packet pairs to single packet and PTP time synchronization.

In summary, the answer to RQ3 is the presented new method of estimating available bandwidth on access links. The analysis and findings in P8, P9 and P10 tie into each other and describe all aspects of the method which have been analyzed.

5. Evaluation and discussion of results

The answer to each of the research questions lead to research contributions, consisting of either a new concept or method. In this section each of the contributions will be discussed against standardisation and related research in the specific field. In Chapter 2 a more general state of the art description was given which applied to my initial view of the research context, while in this section the focus will be only on the contributions.

5.1 Evaluation of Contributions

5.1.1 Contribution 1

A method for controlling the quality levels of DASH in the home gateway

The suggested method of making the home gateway aware of the available quality levels of DASH sessions, and using this as basis for regulating the available bandwidth for each session is related to several IETF standards as well as other research initiatives.

As early as in 1994 the issue of controlled link sharing was discussed in RFC 1633 [38]. The basic functions considered for this purpose was packet scheduling, packet dropping, packet classification and admission control. By combining these functions, certain link sharing objectives could be met by implementing e.g. Weighted Fair Queueing (WFQ). In the same RFC, the first description of Reservation Setup Protocol (RSVP) is given. The discussions on different ways the RSVP protocol could operate lead to a conclusion that resource reservations should be receiver-initiated. It also stated that the receivers are assumed to learn the senders offered flowspecs (i.e. list of QoS parameters) by a higher-level mechanism. Similarities between the RSVP and DASH concepts are present in the sense that the DASH manifest files contains the available flowspecs for a specific video, and the selection among these are done by the client. There are of course significant differences in the sense that DASH does not enforce any reservations along the path, but still the similarities are obvious. In the later RFC 2205 [108] the RSVP protocol changed name to Resource Reservation Protocol but with similar content in this regard.

In RFC 2638 [103], the bandwidth broker concept is introduced as part of a framework for marked traffic allocation. The role of the bandwidth broker (BB) is to control bandwidth shares. However, as opposed to the RSVP approach the BB architecture is in line with the DiffServ [10] idea of pushing complex functionality towards the edge of the network rather than applying it to every node along network paths. The role of a BB is two-fold, whereas the first is to manage bandwidth shares within a certain administrative domain and the second is to manage control message to adjacent domains.

In RFC 2638 the actual size of the administrative domain managed by a BB is not stated, thus it could in theory be a home network. In such a case the home gateway (HG) would be the edge/leaf router responsible for implementing the required functionality. From a customer perspective this makes sense, as the HG is the home network edge towards the Internet. However, from the network operator perspective the HG is not the network edge but rather customer premises equipment (CPE). Thus, the view of the HG as a candidate BB is in line with the BB architecture but not quite in line with the ideas behind the DiffServ architecture.

The allocation requests for resources within a BB domain is according to RFC 2638 either issued by a network administrator, a user or another domain's BB. By including the user as the possible source of an allocation requests, it is a reasonable interpretation of the RFC to say that it also covers the user terminal and applications. Further on, the allocation requests between BB domains may either be dynamical or static. In any case, the policies within each BB domain are independent of each other. It thereby opens up for the scenario of having a single BB domain (i.e. the home network) and the rest of the network to operate on a best-effort basis.

Applying the BB concept on a home network basis and with no inter-BB domain signalling could be considered as a very simple scenario, which does not really deserve to be considered as part of the BB architecture. Some would argue that this scenario is just a matter of QoS support in the CPE. However, as dynamic interactions between user applications and the home gateways for the purpose of communicating resource requirements are not commonly seen, I would argue that this should indeed be considered as a BB scenario. The use of BB in the home network domain has been suggested for the purpose of supporting residential femtocell solutions in [109] [110]. The value of the BB in this context is to dynamically adjust the available capacity for the mobile users entering Femtocell service area. The alternative to apply dynamic control of this would be a static bandwidth reservation on the access link, or an over-provisioning of the access (if possible). In another study [111] the BB concept is suggested as part of a hierarchical admission control scheme for supporting mobile IP usage. In this work, there are two BB domains – addressing the micro and macro level of the involved network. The micro level corresponds to the home network domain.

In the Home Gateway Initiative (HGI) organisation which consists of major service providers and manufacturers of digital home devices, work is put into specifying requirements and test plans for home gateways that support QoS and broadband service delivery in general. In several of their published documents requirements are described which indicate that the home gateway should take a more dynamic approach to how QoS is implemented, and bandwidth allocations in general. In [112] it is stated that in-home bandwidth management is an interesting idea, but with significant technical and commercial challenges associated. For SIP based services in particular, the document [113] describes how a Back-to-Back User Agent (B2BUA) located in the home gateway can be used to intercept the SIP signalling and use this as basis for both admission control and bandwidth management. Such a concept is quite similar to the use of Monitor Plane messages for DASH services in this thesis work. However, there is an important difference in the sense that the B2BUA approach describes a signalling proxy in the home gateway, which requires it to support the specific signalling protocols used. From a lifecycle management perspective this concept is not optimal, as devices behind the home gateway are likely to evolve over time in terms or protocol versions used and supported. In this regard, the use of additional Monitor Plane messages is more attractive as they can use any format independent of DASH control messages between client and server.

Another relevant parallel to the thesis contribution in this domain is the concept of Software Defined Networking (SDN) which has gained a lot of attention both from the research community and the industry. In short, SDN introduces programmability of the network in a more open way than facilitated before. The essence of this lies in that SDN offers a set of external API's which allow an application to program or configure the network. One of the main organisations which are driving the SDN evolution is the Open Networking Foundation. In general, the motivation for SDN is based on the changes in how services are being delivered on the Internet. The strengthened importance of end user devices and also the belief in that a more open network will facilitate innovation in a better way are key drivers for SDN. As for any other new concept or technology which comes around, SDN will represent a contribution in some but not all areas.

Allowing applications installed on end user devices to control network behaviour which could have an impact on others than the specific user and associated organisation is for sure a brave idea. This brings us to the question regarding which functionality is both interesting and recommendable to offer control of across an API to an application. Inside network domains which are under the same autonomous control and the degree of trust is high, the opportunities are greater than in public networks. One of the early deployment examples of SDN of a certain size is the Google Inter-Datacenter WAN implementation using SDN and OpenFlow [114]. As stated by Google, their motivation for this was a need for managing their WAN as a fabric and not as a collection of individual boxes.

Another area where both autonomous control and the presence of trust can be assumed is the home network domain. Such networks are growing in terms of both complexity and amount of devices being served, but they still remain under the control of the household. Another aspect which makes the home network of interest in the context of SDN is that the home gateway node is the key element is terms of giving access to the shared access link. Thus, implementing an API in the home gateway allowing it to receive Monitor Plane messages from DASH clients should be a feasible task.

In summary, the research contribution covering a method for controlling the quality levels of DASH in the home gateway has relevance for on-going developments and standardisation efforts in several areas.

5.1.2 Contribution 2

A method for improving fairness among competing DASH sessions

The suggested method for improving fairness among competing DASH sessions is related to the general topic of inter-flow fairness on the Internet, and to the more specific topic of inter-application fairness in user groups.

A flow is defined in RFC3697 [115] as a sequence of packets sent from a particular source to a particular destination. Thus, by definition - a flow does not necessarily map onto a single transport connection. From an application perspective this aspect is very important as it may utilize several transport connections as part of its operation. With regards to DASH type of services, it is commonly seen [116] [117] that audio and video are carried by separate TCP connections, and for some implementations such as e.g. Netflix [118] there are even multiple TCP connections used for the video part.

Studying fairness on the transport connection level is mainly done for TCP, as these protocols embed functionality for adjusting the rate of a flow based on loss, delay or combination of such events for the purpose of congestion avoidance. The original TCP version as specified in RFC 793 [119] did not contain this type of functionality, but it was introduced in [54] and then included by all later developed versions of the protocol. There are numerous studies published on how TCP flows of the same or different type treat each other in various network scenarios. Each study gives a view on performance issues like fairness which is valid for a specific scenario investigated, but given the dynamic traffic mix on the Internet – one cannot expect to reach a common understanding of which is the best TCP protocol. For video streaming in particular the work in [85] showed that no widely deployed TCP version is able to deliver "best" performance across the range of scenarios studied. In other words, it all depends on the specific scenario which is the best TCP protocol version to use for this purpose.

Following the findings in [75] that TCP was a viable choice of transport protocol for streaming multimedia, several other studies have also supported this. In the study [120] a discrete-time Markov model for TCP-based video streaming is presented, which enables performance evaluation by means of simulations. The presented model gives guidelines for when TCP used for streaming provide satisfactory performance assuming a stop-and-wait playout strategy. However, the applicability of this model for DASH based streaming may be questionable due to their special use of buffering and also playout strategy. It is important to remember that the operation of DASH based services today benefit from the availability of more buffering capabilities, CPU and bandwidth on the client side. These factors make TCP even more suited for video streaming than before, but the flow level fairness challenges remain.

The topic of inter-application fairness in the context of video streaming has not been subject to a similar amount of research as for the flow level case. Keeping in mind that it is the total performance of each application the end users experience, one can easily argue for the importance of this. In an early work [121] the inter-session fairness for layered video services delivered using multicast was investigated. The concept of layered video provides the client with opportunity to select quality level similar to what is possible with DASH, but in this case each quality level is available in the network by its own multicast group. Thus, a technical concept more appropriate for broadcast TV over Internet rather than on-demand content. The two schemes proposed for improving fairness during times of congestion are based on the use of an adaptive join-timer or priority dropping. Their simulation results showed that both methods have potential for improving fairness. Despite the differences in terms of content delivery method used (multicast vs. unicast), the mechanisms investigated could still be used for on-demand DASH type services. The adaptive join timer is actually somewhat similar to the threshold based behaviour outlined in one of the included papers in this thesis (P4).

In another work [122] the topic of quality-fair video streaming from a single video server is investigated. The shared resource in this case is the uplink bandwidth from the video server. It is argued that the fairness achieved on per TCP session level is not enough to guarantee quality-fair streaming to the clients involved. The reasons for this are stated to be differences on the client side in terms of playout rate and playout buffer. The suggested solution to this is for the server to make an estimation of the status for each client playout buffer, and then perform resource sharing based on this. The resource sharing of TCP is overridden by creating multiple TCP connections per client, and adjusting this number up or down whenever the client should receive more or less of the shared resource. The relevance of this method for DASH based services is questionable as the method is likely to be sensitive to TCP version, and the fact that DASH services operate in a per-segment oriented manner. For a multi-TCP solution to be applicable for DASH it would also require the client code to be changed, thus it requires changes both on the client and server side.

The approach taken in [123] includes the aspect that it is not only the bitrate level of a video stream which indicates the actual quality as experienced by the end user. The temporal variations in the video itself, combined with compression schemes targeting specific bitrates will also contribute to potentially noticeable quality variations. Thus, if the fairness of the system is to be engineered from an end-user perspective (QoE), the resource sharing should also take this into account. The paper further discuss average and temporal variations of perceived video quality (PVQ) and which factors that have an impact on this. The definition of PVQ is not very firm, but is used as the QoE metric measuring fair resource sharing. The factors impacting PVQ is further stated to be aspect of wireless channels such as fading and changing network load. Aspects of the time varying nature of video scenes are also mentioned. In summary, the applicability of this work in relation to DASH services is present in the sense that fairness should indeed also include QoE metrics. However, the specific QoE model used in the work can for sure be discussed in terms of validity.

A user-centric approach to fairness among competing video streaming clients in a shared network environment is addressed in [124]. The problems caused by each client operating independent of other devices on the network are emphasized. In order to improve this, an OpenFlow-assisted QoE Fairness Framework (QFF) is suggested. The OpenFlow approach is part of the Software Defined Network concept which has gained a lot of attention over the last years. The suggested framework is relevant for both contributions 1 and 2 of this thesis, as it makes use of the home gateway as the controlling entity for the home network and by this enables fairness control. Instead of using additional Monitor Plane messages as suggested in the thesis paper P3, the use of a Network Inspector (i.e. deep packet inspection) is proposed as part of the QFF. This enables the home gateway to obtain information from the DASH manifest files describing the available quality levels for each active stream. The second part of QFF is the optimization function which combines the collected information and makes decisions on how much resources are to be allocated to each client. The last part of QFF is a notification function which informs each client about which quality levels it should use in order to achieve network-wide QoE fairness. The notification function is required due to the client driven nature of DASH. Comparing this approach to the thesis contributions it is similar in the sense that there is an assumption made regarding relationship between QoE and actual bitrate, and also that the user group of interest is the home network. Significant differences exist in terms of that in order for QFF to function properly the client video players must be changed in order to receive, interpret and act according to the quality level recommendations received from the home gateway.

In [125] techniques for managing the trade-offs between stability, fairness and efficiency when multiple bitrate-adaptive video streaming sessions share a bottleneck are presented. The streaming solutions used in the research are of DASH type, with focus on the Smooth Streaming solution from Microsoft. The challenge of both unfairness and instability during time of congestion is presented. The suggested techniques presented for improving this are called randomized chunk scheduling and stateful bitrate selection. The randomized chunk scheduling technique is very similar to the method presented and analysed in papers P4 [107] and P5 [126] included in this thesis which were published before [125]. The principle used for available bandwidth estimation in the stateful bitrate selection is also quite similar to the model used in paper P4. In summary, findings in [125] support the specific thesis contribution in this domain. However, there is one significant difference in the sense that [125] requires changes to the client player – while in this thesis contribution the suggested method only requires changes on the server side.

In summary, the research contribution covering a method for improving fairness among competing DASH sessions is addressing a topic which is gaining an increasing amount of attention in the research community. There are similarities between the thesis suggested method and work published by others, but the uniqueness lies in terms of simplicity and server side implementation only. The suggested method does not require any advanced algorithms or changes to DASH client and server interactions. Thus, it has a potential of being implemented in existing content delivery platforms.

5.1.3 Contribution 3

A method for achieving shaping effects for traffic aggregates on access link with DASH components.

The suggested method for achieving shaping effects of traffic aggregates on access links with DASH components, relates to both traffic shaping in general and shaping of adaptive video streaming in particular. It does not have any direct relations to functionality subject for standardisation efforts, although it is mentioned in the early DiffServ RFC [10]. Within the scope of DiffServ, a shaper is described as a mechanism used to bring a traffic stream into compliance with a traffic profile.

For the sake of completeness, it is important to mention that traffic shaping serves a purpose also on a higher level than on just individual service and access link level. As described quite well in [127] traffic shaping has positive effect on both the ISP network level and the global Internet backbone level. By using traffic shaping mechanisms the traffic peaks on network links can be reduced and thereby significant cost saving achieved. However, such issues are not related to the contribution of this thesis, and will therefore not be discussed any further.

In the early phases of delivering video over the Internet using MPEG encoded content, it was considered as a service suitable for the Guaranteed Service (GS) class within the IntServ model. The objective of this class was to make sure that firm delay and bandwidth requirements were met. In the work [128] it was shown how a token bucket type traffic shaper applied at the server side could be used to achieve controlled delay for MPEG encoded video carried over UDP. The findings in this early work do not directly apply to the DASH solution of today, as both the transport protocol has changed (TCP instead of UDP) and the adaptive behaviour has been introduced. However, as the cited paper is among the first ones to present a study of MPEG traffic shaping by means of both an analytical model and simulations it serves as a reference point for a lot of later published papers. The most interesting part of the paper with regards to this thesis contribution is the discussion on the burst oriented nature of MPEG traffic – even in the case when it is not delivered as DASH.

One of the more popular video streaming services on the Internet is YouTube. Not only is it being used frequently, but it also represents a significant amount of the traffic volume on the Internet. Thus, it contributes to the characteristics of traffic on an aggregated level. In the work [129] a characterisation of traffic generated by YouTube when accessed from typical end user clients is presented. The characterisation is based on actual measurements of traffic generated on the application layer, and forms the basis for a service model which can be used by either simulators or traffic generators. The streaming solution used by YouTube at the time of the cited research was based on MPEG4 encoding, HTTP/TCP and progressive download (i.e. not DASH). The interesting findings from the cited paper which is relevant for this thesis, is the documentation of traffic patterns dominated by burst components. The burst components are quite strong both in the initial phase and the following throttling phase. What takes place when the stream enters the throttling phase is that the server shapes down the amount of data being passed to TCP to a constant rate. The amount of data already received by the client during the initial phase serves as a buffer which prevents the video stream from halting if the available bandwidth drops below the throttle rate. This type of shaping which occurs at the application level has the benefit of that it easily can relate to the actual bandwidth required by the service. However, it does not address the competition for bandwidth between multiple YouTube sessions being delivered to the same home network.

The bursty nature of YouTube traffic is further discussed in [130] and the challenges this gives in terms of causing packet loss, and making TCP congestion control mechanisms react. In order to improve this, a server side mechanism called Trickle is suggested which uses TCP mechanisms to shape the traffic originating from YouTube content. This is done by enforcing an upper limit for the TCP congestion window as a function of the video streaming rate and round trip time. This approach requires unique TCP settings on a per connection (i.e. socket) basis, something which normally is not directly supported in a standard TCP stack. Setting the maximum TCP congestion window is standard functionality, but this normally applies for all TCP sockets on a server. The applicability of the presented mechanism when DASH is used instead of progressive download is not discussed. However, it is likely that it can still be applied – assuming that the mechanism is able to re-configure the congestion window for a specific TCP socket each time a DASH session changes quality level.

Streaming of video content to mobile devices is also increasing in popularity and volume. In mobile networks the access part have quite different characteristics than in fixed networks, and is considered a much more challenging scenario in terms of being able to guarantee a stable capacity during the lifetime of e.g. a video session. However, such networks also have other types of functionality which can be used for purposes such as traffic shaping. In the work [131] an interesting approach to traffic shaping is presented which utilizes interference control to enforce video streaming rate. The mechanism uses a shaping of the received interference power in order to decrease the variations in throughput for a user by smoothing the transmit power profile of the interfering base stations. Clearly, this mechanism is limited in use to mobile networks but nevertheless very interesting. As stated in the cited paper it requires some degree of quality monitoring and feedback from the clients as input. This is stated to be within the scope of 3GPP DASH specifications [132]. The mechanism does not address the case of shaping traffic aggregates generated by multiple video streaming sessions, but this is natural since mobile broadband access still remains to become popular as the main access technology for serving e.g. a complete household.

The interactions between DASH mechanisms and TCP are interesting in general as they both have the capability of adjusting the bitrate of a video stream. On the DASH level it takes place as part of quality level control, while on TCP level it is part of the congestion control mechanism. It is further also important to emphasize that the specific TCP version used also contributes to how this interplay occurs. A selective manipulation of a traffic flow can be applied in order to trigger certain TCP mechanisms, and thereby achieve a shaping effect of some kind. One such method is called TCP pacing, which is based on spreading the transmission of packets across the duration of an estimated round-trip-time (RTT), rather than sending bursts. This means that packets are spaced at the sender in a controlled way, instead of being sent as fast as possible. The direct effect in terms of shaping is obvious. In the work [133] the effect of TCP pacing is studied for DASH services, with focus on its effect on delay and throughput for streaming sessions. The relevance of this work for the shaping aspect is to investigate the potential side effects of the method – if used for shaping. The findings presented indicate that when pacing is implemented, there are some traffic and delay scenarios where the pacing mechanism gives somewhat lower throughput for the video stream. However, the reductions are not significant. One aspect which is not discussed in the cited paper is how resource demanding an implementation of TCP pacing is for the server. This is a critical aspect if considered applied in CDN platforms serving a high volume of DASH sessions.

There are different objectives for trying to shape traffic in general, and traffic originating from DASH sessions in particular. In the work [134] shaping is applied in order to reduce the problem of quality oscillation for competing DASH sessions. The investigated shaper mechanism is located on the server side, and is focused on reducing the traffic bursts occurring immediately after the client request a new video segment. In an unconstrained mode of operation each new video segment is normally delivered as fast as possible from the server, and then stored in the client playout buffer. The proposed method is based on activating a shaper module when it detects that a DASH session oscillates between different quality levels. When activated, the shaper will limit the throughput for the specific session to a rate slightly higher than the encoding rate of the next segment to be delivered. The reason for setting it higher than the encoding rate it to compensate for communication overhead, to avoid that the playout buffer becomes empty and also to avoid triggering the DASH client mechanism controlling the quality level requests. The proposed method is quite interesting as it selectively takes control of DASH sessions which have unfortunate behaviour, and applies additional control mechanisms on them. However, in order to detect and identify the sessions with unfortunate behaviour, the server side have to perform additional monitoring of all sessions. The complexity in terms of implementing this is not discussed.

The use of bandwidth shaping per DASH session in the home gateway is suggested in [135]. The stated objectives for this are to improve bandwidth sharing accuracy, stability and convergence time. Their suggested method is quite similar to concept described in this thesis included paper P2, which was published earlier. Rather than using Monitor Plane messages as this thesis paper P3 suggest, a deep packet inspection of the DASH manifest files are proposed.

In summary, this thesis research contribution of a method for achieving shaping effects for traffic aggregates on access link with DASH components is focused on the shaping effect on packet level. The other published methods as discussed in this section are mainly focused on shaping effects in terms of bitrate time averages. Thus, shaping effects on a higher level than what was the objective in this thesis contribution.

5.1.4 Contribution 4

A method for estimating available bandwidth on access links when DASH sessions are present.

The suggested method is related to the general topic of available bandwidth estimation, but mainly to the special case of access links where multiple DASH sessions compete for the available bandwidth. The estimation method is based on the use of active probing, rather than passive measurements. The rationale for this approach is the lack of access to the involved network nodes which is typical for an OTT service delivery. The access link scenario represents a special challenge for estimating available bandwidth in the sense that the traffic pattern is very burst oriented, and also that the degree of probing intrusiveness is of concern.

Active probing along network path can be used to detect various events or to estimate specific metrics of interest, whereas available bandwidth is one such metric. The probe traffic to be inserted can be of different types, i.e. in terms of probe packet size, packet rate and time between each packet. What represents a single probe also varies; it could be a single packet, a packet pair, packet quartets or even trains of packets. In an early work [136] the use of packet pairs was suggested as part of a method for detecting link-level loss rates, and in another work [137] the use of packet quartets was used to deduce available bandwidth estimates through end-to-end delay sampling. Following the growing amount of methods for estimating available bandwidth being published, comparison of the different methods from different perspectives became interesting. The classification of methods together with a measurement based comparison of those available was done in [71]. Whether a method is based on self-induced congestion or not, forms the basis for defining it as a Probe Rate Model (PRM) or Probe Gap Model (PGM) type of method. The method suggested in this thesis belongs to the latter category, and does not rely on self-induced congestions.

The general challenges of performing available bandwidth estimations by means of active probing are described in [138]. As stated in this, a common assumption for most available bandwidth estimation techniques is the single-link model with fluid cross traffic. The validity of this assumption can be discussed. The single-link model may not be a bad approach, as the capacity of Internet edge and core in most cases are well dimensioned. Thus, the congestion point is in most cases the access part – and for this a single-link model could apply. The fluid cross traffic assumption is very convenient from an analytical perspective, but in reality this never holds. For an access link in particular, the cross traffic would never be of a fluid type. The typical traffic pattern of today would instead be influenced by applications generating traffic bursts, such as DASH based video streaming. Another relevant challenge presented in [138] is the relation between probing stream duration and time average interval for the estimator. This maps onto the idea behind the suggested method in this thesis paper P10, which uses detection of periodic traffic patterns as an important component. The balance between estimation latency (or speed) and accuracy is also discussed. Knowing that the variance of the sample mean for a probing process decreases with the number of samples, it follows that the accuracy of the result would also follow the same trend. However, with a non-fluid cross traffic pattern this may not hold. Thus, a balance between latency and accuracy is not trivial. In relation to this thesis work, the approach of probing according to strata is meant to address this issue in the sense that each additional sample should contribute to reduce the variance of the sample mean. This is achieved by reducing (ideally avoiding) the amount of probes sent during time intervals when there is no significant cross-traffic. For the sake of clarity, it is important to note that the stratification used in this thesis work is different from the more classical use of this term when describing sampling methodologies [139]. In this thesis work, strata for burst and idle periods are used – while the more common approach is equally sized strata which are not sized or synchronized with the traffic pattern.

In the work [140] the accuracy of available bandwidth estimations based on packet pair sampling is discussed. An analytical model for PGM probing is deduced, which leads to expressions for probing variance. The model includes analytical reflection on how the degree of burstiness in the cross-traffic impacts PGM probing. This model strengthens the motivation for the approach taken in this thesis work, in the sense that probes should be focused on the busy periods – during which the cross-traffic may be close to Constant Bit Rate (CBR). The analysis in [140] of the impact of probe packet size with regards to variance and accuracy may not hold in real life. Their statement is that larger probe packets are better than smaller. The reason for questioning the validity of this statement is that the simulation model used seems to have cross-traffic CBR sources operating at the same link speed as the bottleneck. In reality the cross-traffic will appear containing traffic burst component with bitrates in the order of 10-100 times the bottleneck. The reason for this is that the server side of a video session normally is connected to the network on much higher capacity links than the typical end user access.

The use of large probe packets is also stated in [141] to provide better accuracy than small probe packets in the case of estimating available bandwidth on a residential Internet access. However, the main reason for this conclusion is the extremely high probing rates used in some cases. Such probing rates give the home gateway equipment problems and thereby it will start to drop packets or in other ways influence the traffic flow. In relation to this thesis work, it further strengthens the stratified approach by making the best possible use of probe traffic sent into the network in order to minimize network element impact.

The strong impact of cross-traffic variability on the accuracy for a range of available bandwidth estimation method was documented in [142]. In their comparative study, it was not possible to reach a conclusion with regards to which of the included methods was the best. Reason being that the performance varied depending on the nature of the cross-traffic. These findings also support this thesis approach of including a traffic characterization effort in the probing method.

In the work [143] a similar approach is described which adjust the sampling according to knowledge about traffic type and characteristics. Depending on which network performance metric is being investigated (e.g. loss, delay, jitter) different probing rates and time average intervals are used. The method also includes a mechanism for predicting future behaviour based on estimated behaviour, thus using the output of the probing on a continuous basis as input to the adaptive behaviour. This approach differs from this thesis method in the sense that it relates to cross-traffic service type (e.g. voice, data) and also aims at estimating other parameters than available bandwidth. However, the concept of introducing adaptive probing is quite similar to stratified probing.

Methods for estimating available bandwidth all relate to some interval of time, over which the mean estimator value is computed and presented. Depending on the application of the estimator, the time interval of interest may be different. In the context of DASH services and the potential use of such estimators as input to the client side quality control mechanism, the time intervals of interest would typically be in the order of seconds. This raises the interesting question with regard to which probing method is more suited for estimating time averages. In stochastic systems it is a well know property [144] that Poisson Arrivals See Time Averages (PASTA). Based on this it would be tempting to bridge this property onto active probing methods. However it can be discussed whether the PASTA property then still holds. In [145] it is stated that the PASTA property relates to bias in the sampling process, and not variance. It is also stated that it does not cover bias after a potential inversion process. The role of inversion in this context relates to the process of obtaining the metric of interest based on what has been sampled. The parallel to available bandwidth estimation would be the process of using a sequence of delay samples to construct an estimator for cross-traffic bitrate and thereafter available bandwidth. It is also stated in [145] that the PASTA property does not hold for probe patterns, such as packets pairs or trains of packets. It has only been proven to hold for a stream of Poisson packets.

For applications using available bandwidth estimations as basis for some kind of real-time operation it is important that the method used is able to work according to this. As earlier stated, the variance of the sample mean is reduced by an increasing amount of samples. Thus, a requirement for low estimation latency may come at the expense of accuracy. In a method which belongs to the PRM category (i.e. based on self-induced congestion) which is called BART [72], the use of a Kalman filter is used in order to achieve real-time performance. The role of the Kalman filter is to maintain and continuously update the available bandwidth estimate for each additional sample. The suggested optimization of the BART method is stated to be a tuning of the probing method according to cross-traffic characteristics. This is in line with the contribution of this thesis, i.e. the stratified sampling approach.

In summary, this thesis research contribution consisting of a method for estimating available bandwidth applies in particular for the scenario where we have a single congestion point in the network path of interest and the cross-traffic is dominated by burst components and periodic behaviour. The typical case would be an access link congested by traffic from multiple DASH sessions being delivered to members of the same household. As shown, there is related research by others which are pointing in the same direction in the sense that probing can be done more efficiently if it relates to certain characteristics of the cross-traffic being probed.

5.2 Evaluation of Validity Threats

The research questions addressed by this thesis are all focused on the OTT service delivery model over the Internet. Thus, methods investigated and suggested are based on assumptions on what is possible to achieve in such scenarios. One important aspect in this regard is how interested and active the network providers are in OTT value chains. The assumption made as basis for this thesis research is that the network operators are not active at all which could be considered as a “worst case” scenario. This excludes the opportunity to utilize mechanisms in the network operator domain for OTT service providers, and puts all the attention on service end-point locations. On the customer side this covers the client device, local area network and home gateway device. The OTT service provider side is assumed represented by a CDN node, something which is quite common for popular video services today. If these assumptions do not hold, there are both opportunities for improving the contributions of this thesis and also likely that alternative approaches could be considered.

The first contribution (C1) is in terms of technical depth more a proof of concept rather than an actual method. It is based on the assumptions that OTT service providers are willing to include additional code in their video player clients for the purpose of sending monitor plane messages. The feasibility of this is demonstrated through the thesis, but limited to only one DASH solution. However, similar functionality is most likely possible to achieve also for other DASH solutions. The use of the home gateway (HG) as a controlling entity for the performance of DASH services assumes capabilities of this device, which is not known to be present in any standard HG product in the market today. Using the SDN approach in the HG makes sense from a technical perspective, but it may increase the hardware requirements in such a way that the cost of the device will increase substantially. Further on, the described communication between a client and the HG – in this thesis work described as Monitor Plane messages – must be covered by a communication standard. In light of the simple unidirectional nature of this communication it is likely that instead of a new protocol it could e.g. be included in the DLNA Network Device Interoperability Guidelines (www.dlna.org) as this already has both device and service discovery and control functions.

The contribution (C2) covering a method for improving DASH fairness is only addressing the scenario of multiple sessions being delivered to a user group located on the same home network. It further also assumes that all sessions are either originating from the same server (typically a CDN node) or that some kind of co-ordination between the points of origin is implemented. For scenarios which deviate from this, the relevance of the suggested method must be re-visited. Further on, the method is focused on DASH solutions which are using fixed segment fetch intervals in the order or seconds. This is the case for many DASH solutions, but not all. Thus, the applicability of the method is tightly coupled with this the segment fetch approach used by the specific DASH solution. From an encoding perspective there are good reasons for keeping video segment sizes fixed, but as encoding techniques are evolving this may of course change.

For the contribution (C3) covering a method for achieving traffic shaping effects on packet inter-arrival level - the same assumptions as for C2 are valid, as the underlying mechanism for both methods are similar. In addition to this, C2 also includes an algorithm on the server side which identifies relationships between DASH sessions and specific home networks based on source IP address. This approach is feasible in the typical IPv4 scenario where all users in a home network are accessing the Internet through a gateway with a NAT/PAT function. The sharing of a single public IP address towards the Internet makes it easy for the server side to group DASH sessions according to home network membership. However, as the migration from IPv4 to IPv6 progresses the use of NAT/PAT in home gateways will no longer be needed for the purpose of controlling IP address space usage. In this case, the algorithm for identifying relationships between DASH sessions and home networks must be changed.

The final contribution (C4) of the thesis covering a method of estimating available bandwidth is based on several assumptions which are decisive for its applicability. The approach of stratifying the active probing assumes that the cross-traffic contains periodic components, and also that each period contains sub-periods with different characteristics. In this thesis it is presented that periodic traffic patterns are commonly generated by segment fetch oriented video streaming such as DASH. Combined with the growing popularity of such services, it forms the basis for assuming a presence of cross-traffic with periodic components. Further on, the presence of sub-periods with different characteristics was identified using typical DASH traffic where each segment fetch operation generates a traffic burst followed by an idle sub-period. The presence of burst and idle sub-periods are quite distinct when the cross-traffic is dominated by DASH services. However, the distinction between sub-periods in the cross-traffic is not likely to be this clear when other non-DASH type of traffic also is present. This may increase the challenge of stratifying the probing, but the principle would still apply. As long as the cross-traffic embeds periodic behaviour, and with sub-periods carrying more traffic than others – the amount of information collected from the active probes will increase by pursuing the stratified approach. Thus, although the method is based on several assumptions which may not hold in all scenarios, the principle of stratified probing is likely to have value for many cases where cross-traffic characteristics is possible to obtain.

In summary, for all of the contributions there are assumptions which are important to consider. If something leads to scenarios where these assumptions do not hold, it represents a threat to the applicability and validity of the research findings and suggestions. However, the ideas behind each contribution should still hold and form basis for adjusting the methods according to the new scenarios.

5.3 Reflections on the research context

Service delivery on the Internet according to an Over-The-Top (OTT) model is quite challenging, but at the same time quite appealing from a business perspective – at least for some of the involved actors. A battle between network operators and OTT content providers is caused by their different views on what is fair or not, in terms of who should pay for capacity being used in the involved networks. The network operators would like to collect revenue from both end users for the access part, and from the content providers for allowing them to use capacity in the networks to deliver services to end users. In general, the OTT content providers do not agree with this as they consider the network capacity to be covered by the fee already being paid by end users for the access. This view is supported by the fact that if there had been no OTT content providers present on the Internet, why would end users need an access to the Internet at all ? The third actor in this equation is the global CDN providers, which provides service to the OTT content providers. The CDN providers have OTT content providers as customers, and serve them by accelerating the OTT service performance through a distributed platform. Interesting enough, there is even a battle between the network operators and the CDN providers. Reason being that the CDN providers would like to locate their platform components inside each network, but preferably at minimum cost. The obvious benefits for the network operators in terms of reduced international traffic and improved end user satisfaction do not seem to be enough for the network operators to accept this.

The presence of long lasting battles between different actors on the Internet is the ultimate evidence of that the Internet no longer is driven by a common goal and vision. It also indicates that actors are struggling to establish good ways of working together to evolve the Internet in a direction which serves the best interests of end users. This phenomena is quite well discussed in excellent paper “Tussle in Cyberspace: Defining Tomorrow’s Internet” [146]. In this paper, it is described how different stakeholders on the Internet may have interests which are adverse to each other, and that they tend to favour their own particular interests. However, in some cases the actors interest are not adverse, they are just different. This is referred to as the “the tussle” of Internet which must be addressed in order to secure the evolution of Internet.

In relation to this thesis research one could say that some of the concepts and methods suggested are motivated due to tussle spaces, which are not well enough handled. Investigating methods which can be used for enhancing quality of OTT services without involving the network operator do not come across as the optimal approach. Would it not have been better to involve all actors and seek more optimal methods ? Clearly, the answer to this is yes, but as long the tussle space remains unsolved - the relevance of research in this domain with significant constraints remains.

6. Concluding remarks

Investigating methods for enhancing quality aspects of Internet services in general is a never ending story, and to some extent a chase for a moving target. The target is moving according to changing end user demands and expectations, and is also impacted by the evolutions in technology which enable new opportunities. In addition to this, the dynamics of the Internet in terms of facilitating service delivery to any customer from any provider around the world completes the rather complex picture. Thus, when investigating methods representing potential contributions in this domain one should be somewhat realistic in terms of expectations concerning both impact and duration of applicability for methods suggested.

The scope of this thesis research was set based on observed challenges with popular Internet services of today (i.e. video streaming), and for a quite common end user context in mature broadband markets (i.e. home networks with multiple users). Such a scope may appear narrow at first sight, but as one starts to define use cases it soon becomes evident that even for this case the amount of different scenarios becomes overwhelming. The range of services being accessed on the Internet, the mix of such services on a per user basis and how this combines over time for a user group leads to quite a challenge. For this thesis work, the main service of Interest was video streaming based on the DASH concept, which has grown in popularity over the last years. By setting such a specific scope it was possible to investigate methods for enhancing quality aspects of adaptive video streaming in home networks from different perspectives. The very specific nature of the thesis contributions shows that having this well-defined scope was very effective.

When investigating methods addressing quality improvements of existing services such as DASH, an experimental research approach is both viable and quite strong in terms of verifying the actual effects of the methods. However, it can be quite time consuming to design and implement the required environment to support the different research questions. Throughout the research period leading up to this thesis a home- and access network testbed was used which included both controlled lab and live DASH services. By keeping the experimental environment stable through the whole research period, it enabled the study of different effects for candidate quality enhancement methods. Depending on the topic a hand, different parts of the testbed was utilized – ranging from the client to the server side. By automating the control of multiple video streaming clients, the network capacity and also the video content profiles – repetitive experiments could be run in an efficient way, thus providing enough result for a statistical analysis.

The choice of service delivery model of interest for the thesis research (i.e. Over-The-Top), gave both directions and limitations in terms of what should be investigated. In this scenario the active involvement of the network operator cannot be assumed, and therefore focus should be put on the end-points of the service (i.e. client and server) and potential opportunities there. As discussed in Section 5.3, such constraints may lead to sub-optimal research approaches and methods. However, as long this is the typical case for OTT services it will still remain an interesting research topic. Applying intelligence in service end-points and at the edge of the involved networks, are also in line with the DiffServ architecture. In addition to this, the issue of scalability also speaks in favour of investigating methods which are based on service end-points, rather than only the network part.

From a pure technical perspective, one could wish for that network operators and OTT service providers managed to overcome their to some extent self-centric approach to Internet based service delivery. The use of the term self-centric in this context relates to the observed reluctance to co-operation between network operators and OTT service providers. Both actors should acknowledge that each of them cannot live without the other. No services, no need for a network and without the network – not possible to deliver services. With this in mind, a basis for co-operation should be present.

Assuming that network operators and OTT service providers managed to start working together and investigate how they together could provide better services to Internet users in an efficient way, a wide range of interesting opportunities appears. My personal belief which has grown through my research period is that the potential in using new technical knowledge components as basis for service production carries great promise. The release of such components from both parties to the other may trigger brand new ideas at both sides. This could also as a secondary effect lead to new business models which could further stimulate the co-operation. As an example in this regard the geolocation service offered and used on the Internet could be considered. This type of information is easily available for the network operators, but since they have not managed to utilize it in service production themselves – they are in most cases not making it available to anybody. As a result, a whole range of rather in-accurate geolocation services exist and are being used today on the Internet. The appearance of such services is typically for the Internet. Whenever there is a demand for something, it will be provided – by someone – but not always with a high degree of quality. It would have been better if this type of information was made available by the network operators across a real-time interface and potentially as a payable service.

There are also similar examples when it comes to Over-The-Top video streaming services. Knowing basic things like the access capacity of a specific user represents value as it enables things like personalization of content being promoted (e.g. avoid promoting HD quality content to low bandwidth customers), and also initialization of the correct quality range for DASH services. In relation to this thesis work in the area of available bandwidth estimation, knowing the access capacity for a specific customer is a pre-requisite for a PGM type of method. Again, this type of information is easily available on the network operator side, but has to be probed or benchmarked by the OTT provider if not made available by the network operator.

From the network operator side, there is also information which is interesting to obtain from the OTT provider. Identifying users which appreciate video services could be used as basis for both upselling the access product to the specific customers and also as basis for facilitating CDN nodes in the network. It could also be used as input to network design in the sense that a more timely capacity planning can be achieved, and also a better (technically and commercially) selection of network peering partners nationally and internationally can be achieved. Further on, having knowledge about things like where the next major live video streaming events will originate from is very useful for the network operators as it would give them the opportunity to prepare for it. Such preparations would reduce the chances of congestions in their network and thereby avoid customers experiencing low quality of all services being accessed.

Looking further into the future, it is interesting to reflect how the Internet will evolve from an architectural perspective. The success of the Internet has been based on its initially simple and evolutionary approach to whatever new functionality was required. Combining this with the active involvement of a non-profit Internet community, new services have appeared much faster than the traditional telcos may have appreciated from a business protection perspective. However, whether this really can continue is an interesting question. Is the Internet architecture the right choice for the future, and if not – how can it be changed ? The answer to this is not clear, but there are discussions on-going which outline alternative paths of evolution.

Among the more radical answers to the future Internet architecture is found in the work [147] where it is stated that the well-known layered TCP/IP architecture is the wrong approach. The alternative presented is based on the principle that networking should be viewed as Inter-Process Communication (IPC). It differs from the current TCP/IP architecture in the sense that networks are not considered as a layered set of different functions, but instead a single layer of distributed inter-process communication. In this model, there are mainly two type of entities – application processes and distributed IPC facilities.

The less radical answer to the future of Internet may be based on the ideas behind Software Defined Networking (SDN). The motivation behind SDN is to improve the control plane of the current Internet, while keeping the data forwarding plane unchanged. The changes in the control plane is facilitated by opening up API's toward the controller part of the network components, enabling the applications to take more control of network behaviour. For sure this will open up new opportunities in terms of how services can be delivered to end users across the Internet, and especially for the more demanding services such as video streaming.

However, it is not only a question of how the Internet will evolve as a network, but just as much how the services will change. Foreseeing the results of a creative Internet community is not possible, thus unexpected services and also service delivery models are likely to appear. One of the facilitators for this might be the migration to fibre based Internet access with symmetrical capacity up to gigabit levels. As this takes place, each customer site becomes a viable point of service origination – also for capacity demanding services. Already existing or new peer-to-peer (P2P) architectures for service delivery may then become an excellent choice even for commercial services. As discussed in Section 2.2 of this thesis, the use of P2P technology in CDN platforms is already being considered.

The importance of providing services of high quality is easy to acknowledge, and has been the mantra for the telecommunication operators from the beginning. The migration of services onto IP based networks, and the Internet in particular – has in my view changed how all actors relate to service quality. My personal opinion on this is that we have become less concerned about the general quality levels for services and instead more aware of how individual users experience the service. Something which supports this statement is the rapid growth of video streaming services on the Internet, carried by the involved networks as part of the best-effort traffic class. The availability of video content which matches each individual's personal interests (e.g. YouTube) is still growing in popularity even though the absolute quality levels of the streams are unpredictable. This shows that if end users appreciate certain aspects of a service (e.g. personalized content, terminal flexibility etc.) they are willing to accept "good enough" quality levels from the network side. Based on this, it is my belief that future successful services should be developed with end user perception of quality in mind. As end user perception is a complex issue, and not really a traditional engineering discipline – future successful service developments should become a more multi-disciplinary process.

What concerns the contributions of this thesis, my expectations are that some of them should be of interest for CDN technology providers. The beauty of the suggested method for enhancing fairness and to achieve traffic shaping effects, lies in its simplicity and thereby also feasibility of implementation - without any performance degradation for the CDN nodes involved. Further on, the stratified probing method for available bandwidth estimation introduces a new way of thinking in this domain. As already stated, the method has most potential if combined with other methods, but the principle of aligning an active probing according to cross-traffic profiles is something I hope will be subject for further research and eventually real-life use for some purpose.

Part II: Included Papers

PAPER 1: Monitoring and Control of QoE in Media Streams using the Click Software Router

Bjørn J. Villa, Poul E. Heegaard

Published in
Proceedings of Norsk Informatikkonferanse 2010 (NIK 2010)
Tapir Akademiske Forlag, ISBN 978-82-519-2702-4. pp. 24-33
Nov 22-24, 2010, Gjøvik, Norway

Monitoring and Control of QoE in Media Streams using the Click Software Router

Bjørn J. Villa, Poul E. Heegaard

Abstract: Services provided Over-The-Top to Internet users are becoming more advanced and the business models for operating in this domain are becoming more sustainable. When services of this type are something which is paid for, the importance of optimizing Quality of Experience (QoE) is crucial. In this paper we study and test a selected mechanism PostACK for controlling TCP flows, and implement a home gateway testbed using the Click Modular Router for performing live experiments and measurements. The analysis and measurements show that it is highly likely that TCP mechanisms can be used to control the QoE of these services, assuming that there is a known user preference. One of the media streams investigated was an adaptive http streaming solution based on Microsoft Silverlight technology.

1. Introduction

There has been ongoing research in the area of adaptive networks and applications for decades, in the telecom industry in general – and in the Internet Community in particular. With the success of broadband Internet over the last 10 years, these capabilities have contributed to a rapid service development on top of the Internet infrastructure. However, the majority of these services have been free and therefore the user expectations concerning quality have been moderate.

With an increasing range of commercial services delivered Over The Top (OTT) to Internet users, more focus has been put on the Quality of Experience (QoE) [15] in order to ensure successful and sustainable business models. The main characteristic of an OTT service is that the network operator is not actively participating in the service production except for transporting it as part of the best effort Internet service.

The objectives of the work reported in this paper were to investigate and test selected mechanisms for QoE monitoring and control in media streams, with main focus on emerging solutions for adaptive streaming. Further on, a basic evaluation of Click [9] as a platform for performing this type of research was important.

The structure of this paper is as follows. Section 2 introduces a framework for QoE optimization; Section 3 describes the potential role of TCP rate control in QoE context; Section 4 describes the home gateway testbed used in the experiments; Section 5 presents measurement results and analysis; Section 6 presents the conclusions and Section 7 outlines future work.

2. Framework for QoE optimization

The addition of a Knowledge Plane in network architecture as an addition to the well-known control and management plane was originally proposed by the authors of [4] in 2003. The purpose of this Knowledge Plane was to give a unified view of network aspects, to analyze it – to explain it – and finally also to make suggestions on what to do in order to achieve specified objectives.

Following the idea of knowledge as key component in order to achieve automation with a high degree of quality and precision, IBM published in 2003 The Vision of Autonomic Computing [8]. Although networking was not specifically addressed, it still applies as a good model for this and extends the ideas and concepts for the subset given in [4]. IBM defined the concept of an autonomic manager which had knowledge as the centre point, with a loop of operations around it (monitor, analyze, plan, execute).

The use of a Knowledge Plane (KP) in the networking context, and the ideas from autonomic computing [8] was picked up by the MUSE Project "Advanced features for MM enabled access platform" [5] and formed the basis for a lot of their work. In addition to the Knowledge Plane, the referred MUSE project and following work [1] [2] [3] from the same research groups – lead to the proposal of having Monitor Plane (MP) and Action Plane (AP) components in different parts of a network.

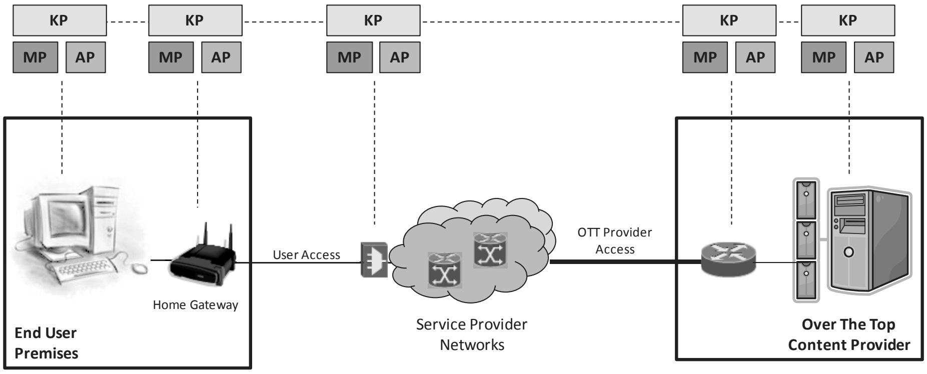


Figure 1 Reference model for QoE optimization research

The framework used for Monitoring and Control of QoE in media streams in this paper is based on the referred work by others, but with primary focus on potential MP/KP/AP functions in the home gateway. The illustration in Figure 1 describes the framework in a somewhat broader scope, which will be used for the continuing research carried out by the authors of this paper. To ease the understanding of these components, the following definitions are considered useful

Monitor Plane (MP): Measurements of different aspects of a service or an element.

Knowledge Plane (KP): The composition and reasoning of collected and exchanged information

Action Plane (AP): The execution of operations aimed at improving QoE

3. TCP Rate Control

Published statistics [10] [11] together with basic knowledge about how dominant Internet based applications work, tells us that TCP is by far the dominant transport protocol. Although the numbers are varying from network to network, they typically report between 75-85% of TCP traffic and the remainder being UDP (Source: Uninett, Spring 2010).

The transport protocol TCP is recognized as a well performing protocol in order to adapt to different network conditions. TCP tuning methods for different networks and applications have been thoroughly analyzed in literature, but there are still more work of scientific interest in this domain.

The emerging demand and volume of real-time applications using Internet have resulted in enhancements of TCP and associated protocols in order to make it more suitable for this type of applications. The work has resulted in both IETF announced protocols such as RTP (RFC 3550) and RTCP (RFC 4961), but also more proprietary protocols such as RTMP (Adobe).

Not all of the TCP extensions have gained the same success. Leading equipment and application vendors have chosen to put potentially enhanced transport layer functionality into the higher protocol layers instead of using e.g RTCP. This is the case for adaptive streaming in Microsoft Silverlight based platforms. The approach taken by these vendors are based on http tunneling/transport, which hides whatever they need of additional functions inside the http payload.

In light of this, in order to influence TCP traffic for QoE optimization purposes – one should work with methods related to the basic TCP mechanisms and behavior and not assume too much about the presence of recent additions.

In Figure 2 an illustration is given on how a highly efficient TCP based mechanism could be used to optimize QoE for selected and preferred content, at the expense of other traffic. By knowing the user preferences and at the same time being able to recognize the types of traffic, different kinds of QoE optimizer schemes would be possible.

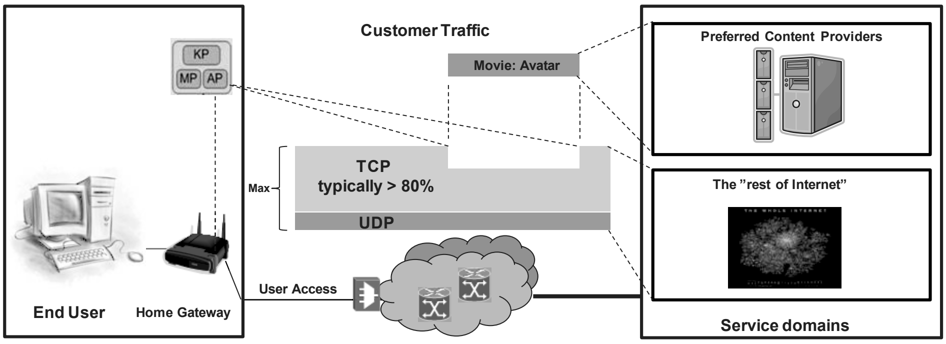


Figure 2 TCP Rate Control as AP component in Home Gateway

Although tempting to think of TCP as "just TCP" there are different versions used even of this protocol [14]. The list of versions is long starting with the rather well known and somewhat old Reno and Tahoe, but also including the more recent TCP version aimed at higher speed networks HS-TCP (RFC 3649). Identifying which TCP version is being used by a certain application or flow is unfortunately far from trivial since there is no field in e.g the TCP header used to identify this. As a result, there is a significant amount of research activities in the field of performing real time identification of TCP version [12]. This is outside the scope of this paper, but still important to be aware of.

There are numerous approaches available for optimizing or in some way regulating TCP flows, but they all relate in one way or another to the basic flow control concept in TCP – where window size and acknowledgements are key. Some examples of mechanisms are controlled drops [6], TCP pacing [7], PostACK [13], ECN (Explicit Congestion Notification) and endpoint window size tuning. One could easily argue that in order for any of these to be accurate, they would need the support of real time TCP version identification [12]. However, it is also interesting to see if one can make such TCP rate control mechanisms itself adaptive, and maybe even introduce some learning capabilities and in this way reduce the required flow level analysis.

In the work reported in this paper a concept derived from PostACK [13] have been chosen as a starting point for studying how and if adaptive media streams such as those generated by the Microsoft Silverlight solution can be regulated and/or controlled. The idea is to see how these adaptive media streams are affected, and compare this to the more traditional service types and for that purpose FTP file download was chosen.

In brief, the concept of PostACK is to introduce a variable delay component for the TCK ACK messages sent from the client to the server, and in this way make the server slow down the sending rate.

4. Home Gateway Testbed

The Click software router [9] is a solution for building your own experimental router or just to analyze traffic. It can be run on almost any industry standard PC type of platform with multiple interfaces and use various types of Linux OS. As part of the SW distribution there is rather rich library of features and functions which can be used, and it is also possible to build your own functions by programming in C++ and integrating this with Click.

Click can be run both in what is called User Level and Kernel Level. In the first mode it could be considered as just a low level packet sniffer, which is highly programmable – but at the same time slightly more demanding to use than solutions like Wireshark. In the second mode, it takes over the kernel routing and interface packet handling completely – and is then capable of doing the role of a regular router.

The main reason for using Click as the key testbed component was the flexibility of this solution, and also that other relevant projects [5] had been using it with good results. Further on, as communication with other entities is foreseen required for future research, using an open source platform based on standard Linux OS was considered important.

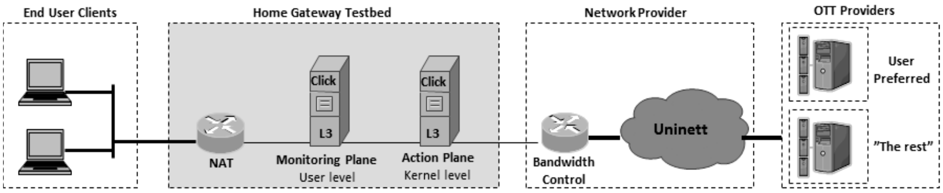


Figure 3 Home Gateway Testbed

In the Home Gateway Testbed established as illustrated in Figure 3, a total of four components are included. There are two separate Click installations, one used as Monitoring Plane (Click Userlevel) and the other as Action Plane (Click Kernel level). Although possible to combine, it has clear benefits to split Click Userlevel and Kernel level operations, in order to get the best out of both. This structure also has a value for future work when KP/AP/MP communication will be prototyped.

In addition to the Click installations, there are two industry standard routers involved – one which has the role of setting the available bandwidth according the desired levels (10Mbps, 5Mbps and 3.5Mbps) for the measurements to be done and the second for the sake of performing Network Address Translation (NAT). These functions could also have been moved into the Click, but in order to keep complexity low and maintain focus only on the issues at hand, they are preferred kept separate.

The Home Gateway Testbed as implemented is a fully functionally solution, with characteristics similar to Fiber-To-The-Home (FTTH).

5. Measurement Results

Measurements of PostACK impact on media streams have been done for FTP file download and a HTTP based adaptive streaming service from www.tv2sumo.no. The common aspects of these two services are that they are both TCP based and also that they typically generate a significant amount of traffic (>1Mbps). The difference between them lies of course in the basic protocol behavior, but more important in this context is that the HTTP service has some target delivery rates. This set of target delivery rates represents the granularity of the service adaptivity. The relevant levels for the measurements reported in this paper were 300Kbps, 700Kbps and 1300Kbps.

The measurements have been done in a live network, involving multiple network operators – Uninett on the access side and a commercial operator serving the content provider involved. For the adaptive streaming service, the TV2Sumo solution was used. The adaptivity of the TV2Sumo service operates at 2 sec intervals, which would be the target delivery rate change frequency.

The measurements are done on IP level for both services and therefore reflect the IP level bitrate, and not the application (HTTP or FTP) payload level bitrate. Measurements were done for three different access capacities (3.5Mbps, 5Mbps and 10Mbps) and the PostACK value used was from 0 to 300ms in 50ms increments. The PostACK was imposed for 100 seconds, always turned "on" at t=50 and turned "off" at t=150. Measurement period was 300 seconds in all cases, starting from a stable situation. The bitrate measurements were done in discrete 1 sec and 5 sec intervals.

As illustrated in Figure 4 the PostACK function is located in the Action Plane, introducing delay for selected TCP ACK packets, while the rate measurement function is located in the Monitoring Plane operating at the TCP data packets.

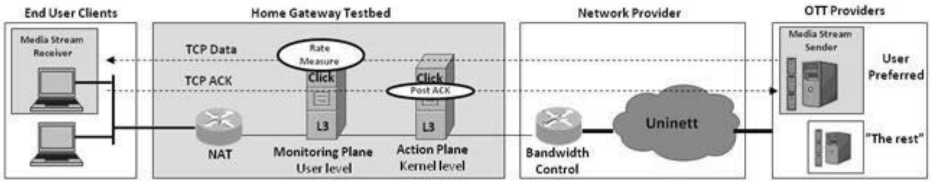


Figure 4 Location of PostACK and Rate measurement functions

All measurements have been repeated between 10-15 times, during different time of day and day of week. The results were all similar to what will be presented in the following sections. The measurements have been done in sequence, and not in parallel.

5.1 PostACK impact on HTTP streaming

The measurements presented in Figure 5 are from the scenario with an access capacity of 3.5Mbps and where the TV2Sumo service is running in adaptive mode, with an initial delivery quality of 1300Kbps. The measurements presented are using the 5 sec discrete intervals.

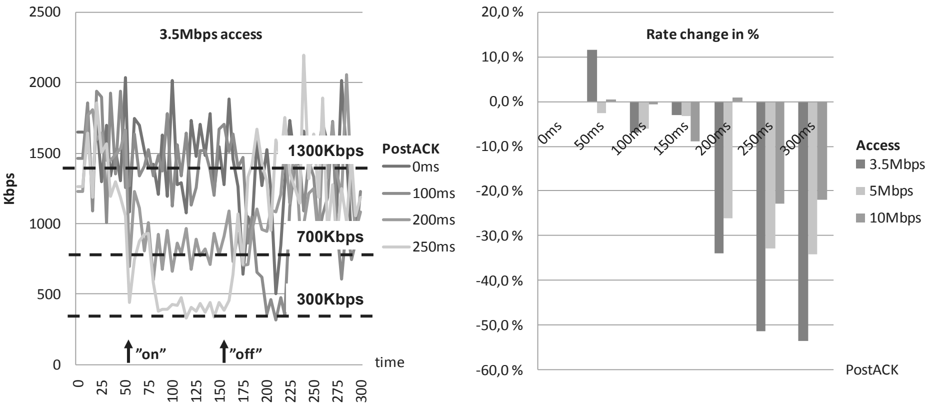


Figure 5 PostACK effect on Adaptive HTTP Streaming

The first observation which is made is the rather high degree of burstiness in the traffic generated even though the target rate is fixed. This is most likely because the client/server application at hand is utilizing a somewhat aggressive buffering scheme.

For the different PostACK values, we can see that the levels of 200 and 250ms make the content server reduce the bitrate and quality. This observation is correlated with a visual inspection of what the content player on the PC is reporting as quality level. The PostACK level of 200ms brings the target quality down to 700Kbps, while the PostACK level of 250ms brings the target quality down to the lowest level – being 300Kbps. The bar graph on the right side illustrates the rate changes for more PostACK values and different access speeds, which clearly indicate that the PostACK impact depends on both the delay component introduced and the access speed.

5.2 PostACK impact on FTP download

The measurements presented in Figure 6 are from the scenario with an access capacity of 3.5Mbps and the FTP file download service is running at a level well below its observed maximum capacity which was observed to be in excess of 10Mbps. The measurements presented are using the 5 sec discrete intervals.

The first observation is that the PostACK levels below 150ms actually gives an increased rate. This may indicate that the additional delay made the TCP window sizing mechanism end up at a more optimal level with the additional delay present, or that the session has entered into a mode with excessive overhead. The increase is only observed at the 3.5Mbps access capacity, while we on the 5Mbps and 10Mbps access measurements instead see a drop in rate for these PostACK values

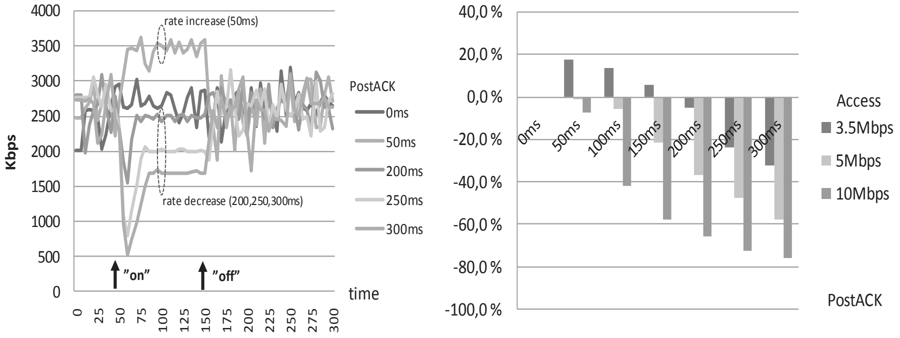


Figure 6 PostACK effect on FTP download

For the PostACK values of 200ms and above we observe different levels of reduced IP level download rate, and we also observe the distinct TCP window decrease/increase phase immediately after the hard PostACK "on" action.

As can been seen from the rightmost graph, the actual effect of a certain PostACK value is also related to the access capacity – and not only the value itself.

5.3 Analyzing the results

It is important to state that the measurements are to be considered as samples, as they have not been repeated enough times in order to be subject to a statistical validity check.

However, as the series of measurements done always have given the same result, it is a strong indication on the effect of the imposed actions.

The values of PostACK which caused an effect in the TCP rates were considered as somewhat high in the beginning, as the author had expected an effect at lower PostACK values. The original expectation was that almost any delay in the ACKs would slow down the TCP sender. However, thinking more closely about this issue one will realize that adding a constant delay component will not change the pace of ACKs, except for at the specific time when the delay component is added – and since a TCP sender basically is "clocked" by the reception of ACKs the sending rate would not really need to change. However, there is a limit for how long a TCP sender is willing to wait for an ACK before it considers the respective data segments as lost – and this limit is known as the TCP-RTO (Retransmission Timeout). The TCP policy concerning use of this timer is governed by RFC 2988, but even in this field there are room for adjustments and optimizations for different purposes. As a result, different operating systems and associated services tend to use different values for TCP RTO.

The thresholds for PostACK values which will cause an effect in TCP sender rate is actually the value which makes the sender detect that ACK's are not received before the expiration of TCP RTO and therefore data segments are considered as lost. This event will trigger the specific TCP version in use to perform congestion control, and the new TCP rate would potentially stabilize according to the new RWIN/RTT ratio, if not affected by other aspects as well of course (e.g buffer issues causing drops). In the table below, the calculation is shown, which is in line with the measurement in Figure 5, for the three cases where PostACK caused a decreased rate.

$RWIN_{byte}$	$RWIN_{bits}$	RTT_1	BW_{Mbps}	RTT_2 (=RTT_1+PostACK)	$RWIN_{bits}/RTT_2$
65340	522720	~5ms	3,5	205ms	~2,5Mbps
65340	522720	~5ms	3,5	255ms	~2,0Mbps
65340	522720	~5ms	3,5	305ms	~1,7Mbps

Table 1 – RWIN/RTT calculation after PostACK for FTP

In summary, the way FTP is affected by PostACK is first that a window reset is triggered due to congestion control initiation, and thereafter the rate grows up to the new RWINbits/RTT2 level.

Knowing the exact threshold for which delay value would make TCP RTO expire, requires in depth knowledge about the operating systems and also current network conditions. In the RFC 2988 formulas are given in order to calculate the recommended initial and subsequent values. Using these formulas for the FTP service and measurements observed it is likely that the TCP-RTO was in the order of 30ms.

For the adaptive HTTP streaming service, the same effect is not visible in the graph as the HTTP client also gets involved as a resulted of the imposed PostACK event. The triggering of TCP congestion control is noticed by the HTTP client as frame drops on mpeg level, and therefore it requests a lower quality of the content currently viewed from the server. Note that this change (adaptivity) is actually initiated by the HTTP client. However, the PostACK effect caused a change in quality level and this was exactly the purpose. From comparing the FTP and HTTP measurements, it is reasonable to expect differences in TCP version and also TCP-RTO value.

6. Conclusion

The objectives were to investigate and test selected mechanisms for QoE monitoring and control in media streams, and the selected services generating such streams were an adaptive HTTP based streaming service from TVSumo and a regular FTP file download service. The analysis and measurements show that it is highly likely that TCP mechanisms can be used to control the QoE of these services, assuming that there is a known user preference. The chosen PostACK mechanism did have a regulating effect on both services, independent of access rate studied – although the results varied.

The specific PostACK effects on the selected media streams is at this time only to be considered as indicative, as more thorough analysis is required – and also more measurements of the same type. Therefore one cannot conclude at his time that the PostACK mechanism could obtain a fully controlling effect, but rather that the results are promising enough to justify more effort to be spent on it. In addition to PostACK, there are also other ways of regulating TCP flows, such as controlled drops - which could be considered as a supplement.

The experience gained by using Click as both as a Monitor Plane and Action Plane in a home gateway testbed has been very useful, and the platform is considered as a very good choice for performing future research in the same or related fields. However, it should be noted that when using Click for measurements it is recommended to have access to a more standard packet sniffer tools as well, such as WireShark. The purpose of this would be to reduced the chances of overlooking something when you develop the Monitoring Plane in Click.

7. Future Work

Following the framework for QoE optimization as described in this paper there are obviously much more work to be done in this field.

In order to strengthen the initial results as reported, the PostACK function itself should be made more flexible. One thing which could be done is to make it gradually introduce and subsequently remove delay components and being able to control this by external signals. Achieving such adaptivity would make it easier to cope with unknown factors related to TCP versions used and timer settings in involved operating systems. There is also a need to study other mechanisms such as controlled dropping, as an alternative or supplement to PostACK. How to control non-responsive flows (UDP) is of course also an issue which needs to be addressed, and the addition of multiple flows would also be an important issue to address.

The effect on adaptive http streaming from TCP rate control mechanisms is considered as the most interesting one to invest more research effort into due to the age of this technology together with its promising capabilities. Understanding the behavior in more detail is crucial in order to make good progress in terms of improving the QoE for the services. Using the received mpeg frame rates at the client side in real time and knowing the threshold for when quality level change will be requested could be very useful.

It is a challenge to know which services are preferred by the user, in order for the QoE optimization mechanisms to know what to actually optimize. The first step in this direction would be to facilitate Knowledge Plane interaction between the components involved, in order to access user preference information located in this distributed plane.

8. Acknowledgements

The reported work is done as part of the PhD studies for the first author which is an integrated part of the Road to media-aware user-Dependant self-aDaptive NETWORKS - R2D2 project. This project is funded by The Research Council of Norway.

References

[1] Latr´e, S., Simoens, P., De Vleeschauwer, B., Van de Meerssche, et.al: An autonomic architecture for optimizing QoE in multimedia access networks. Computer Networks 53(10), 1587–1602 (2009)

[2] Steven Latr´e, Stijn Verstichel, Bart De Vleeschauwer, et.al: On the Design of an Architecture for Partitioned Knowledge Management in Autonomic Multimedia Access and Aggregation Networks. MACE 2009

[3] Steven Latr´e, Pieter Simoens, Bart De Vleeschauwer, et.al: Design for a generic knowledge base for autonomic QoE optimization in multimedia access networks. Network Operations and Management Symposium Workshops, 2008. NOMS Workshops 2008. IEEE

[4] David D. Clark, Craig Partridge, J. Christopher Ramming† and John T. Wroclawski: A Knowledge Plane for the Internet. SIGCOMM’03, August 25–29, 2003

[5] MUSE Project Deliverable D B1.8 (public version): Advanced features for MM enabled access platform, 2006 (http://www.ist-muse.org/deliverables_list.htm).

[6] Hani Jamjoom, Kang G. Shin: Persistent Dropping: An Efficient Control of Traffic Aggregates. SIGCOMM’03, August 25–29, 2003

[7] Amit Aggarwal, Stefan Savage, Thomas Anderson: Understanding the Performance of TCP Pacing. Proc. IEEE INFOCOM 00, Tel Aviv, Israel, 2000.

[8] Jeffrey O.Kephart, David M. Chess: The Vision of Autonomic Computing, 2003

[9] E. Kohler et al.: The Click Modular Router, 2000

[10] Marina Fomenkov, Ken Keys, David Moore, and k claffy: Longitudinal study of Internet traffic in 1998-2003

[11] Wolfgang John and Sven Tafvelin: Analysis of Internet Backbone Traffic and Header Anomalies observed. IMC’07, August 24–26, 2007.

[12] Junpei Oshio, Shingo Ata and Ikuo Oka: Real-Time Identification of Different TCP Versions. Lecture Notes in Computer Science. Springer Berlin / Heidelberg

[13] Huan-Yun Wei, Shih-Chiang Tsao, Ying-Dar Lin; Assessing and Improving TCP Rate Shaping over Edge Gateways. IEEE Trans on Computers, March 2004

[14] Bogdan Moraru, Flavius Copaciu, Gabriel Lazar, Virgil Dobrota: Practical Analysis of TCP Implementations: Tahoe, Reno, NewReno. RoEduNet International Conference, 2003

[15] E. Areizaga, L. Pérez, C. Verikoukis, N. Zorba, E. Jacob and P. Ödling, “A Road to Media-aware User-dependent Self-adaptive Networks”, IEEE-BMSB, Bilbao, Spain, May 2009.

PAPER 2: Towards Knowledge-driven QoE Optimization in Home Gateways.

Bjørn J. Villa, Poul E. Heegaard

Published in

Proceedings of The Seventh International Conference on Networking and Services 2011 (ICNS 2011)

ThinkMind Digital Library, ISBN 978-1-61208-133-5. pp. 252-256

May 22-27, 2011, Venice, Italy

Towards Knowledge-driven QoE Optimization in Home Gateways

Bjørn J. Villa, Poul E. Heegaard

Abstract: This paper presents the concept of using distributed knowledge components as basis for a Quality of Experience optimization process. We also present simulation results indicating the potential in using this approach for access and home networks. The main novelty of the paper is the presentation of how specific end user preference information can be combined with specific content provider service information, and used as input to an optimization process in a home gateway device. The results show that the effect of doing this is significant.

Keywords: QoE, Home Gateway, Adaptive Services

1. Introduction

The focus on QoE (Quality of Experience) rather than just QoS (Quality of Service) has been growing in strength over the last years. The main reason for this relates to the acknowledgement of that users are not equal. The QoE approach covers not only technical metrics, but also metrics describing the uniqueness of a specific user (cf. Table 1). As a result, it represents a measure of the overall customer satisfaction with a service or vendor. This makes it more suitable for user oriented service delivery architectures [3].

Table 1. Example QoS and QoE metrics

QoS metrics	QoE metrics
Bandwidth	Perception
Delay	Preferences
Packet Loss	Expectations
Jitter	Acceptance
Availability	Price

The traditional approach of assigning a fixed priority per service or service class, and then implement a QoS design may not be rich enough to support more advanced QoE optimization schemes. Even with the full range of DiffServ values/classes [12], this will be a limiting resource. Further on, the actual QoS implementation with a high number of classes would have significant complexity issues. As an alternative to this, the concept of knowledge based QoE optimization is proposed.

For content providers operating in the Over-The-Top domain it is natural to focus more on the QoE dimension rather than the QoS subset, as the latter would be partly outside of their control. In line with this statement it is easy to understand that this type of content providers would appreciate techniques enabling them to adapt their service delivery according to different users and varying network conditions. Further on, the location of effective optimization processes outside of the network operator domain is beneficial, as this would not require involvement from the network operator.

The structure of this paper is as follows. Section 2 provides an overview of state of the art in the relevant field and also defines the objectives of the research reported in this paper; Section 3 describes the role and components of the Knowledge Plane; Section 4 describes the simulation model; Section 5 presents simulation results; Section 6 presents an analysis of the results; Section 7 provides the conclusions and an outline of future work.

2. State of the art

The framework used for QoE optimization in a home network environment is in line with related work as stated in [9][10][11] and illustrated in Figure 1. The addition of a Knowledge Plane in network architecture as an addition to the well-known control and management plane was originally proposed by the authors of [5]. The purpose of this Knowledge Plane was to give a unified view of network aspects, to analyze it – to explain it – and finally also to make suggestions on what to do in order to achieve specified objectives. The use of a Knowledge Plane in the networking context, and the ideas from autonomic computing [8] was taken further by the MUSE Project "Advanced features for MM enabled access platform" [6]. Their work lead to a proposal of having Monitor Plane (MP) and Action Plane components distributed across a network, including the end systems.

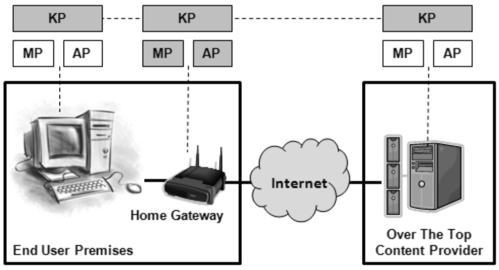


Figure 1. Optimization framework

The main difference between the optimization model used in this paper and earlier work by others is the inclusion of KP/MP/AP components from end user and content provider domains (cf. Figure 1). The KP components from these domains are used as input to an optimization process in the home gateway which then is studied in this paper.

The content of the Monitor Plane and Action Plane is not the main focus of this paper, as we just assume their presence in the home gateway. More information on this can be found in previous work [9][15][17][18][19]. The type of Action Plane components applied would to a large extent depend on whether the traffic flows subject to control are of a responsive (TCP) or non-responsive (UDP) type. Related work in this area can be found in [1][7][20]. It is also important to note that the location of Action Plane components in the home gateway and not at network edge impose some challenges. Reason being that the congestion point for downstream traffic is at network edge.

The objective of the research documented in this paper is to support the statement that QoE optimization mechanisms for Internet services can be implemented in the home network domain, with the use of appropriate knowledge sources. The chosen method for providing this support is by means of simulation of a defined service usage scenario, with variable input parameters.

3. Knowledge Plane

The Knowledge Plane is represented by information objects distributed across the platform components involved.

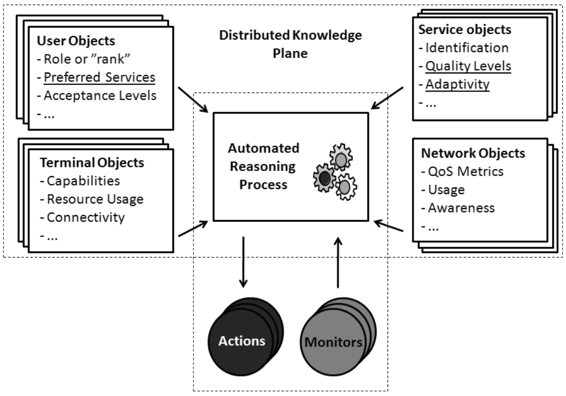


Figure 2. Knowledge Plane input to reasoning process

The use of Knowledge components in an optimization process requires a reasoning process (cf. Figure 2). This reasoning process combines and interprets the different components, allowing them to be used for some actions, and also effects to be monitored and understood.

The information objects used in the work reported in this paper are the user and service objects, and selected parameters from these (user: preferred service, service: quality level and adaptivity)

3.1 User Objects

The list of user preferences and associated capabilities which, could be used as input to an optimization scheme is potentially long, and depends to a large extent of the type of users being discussed (residential vs. business). What is considered as important by one user may not be of interest at all to another user, and vice versa. The thresholds for what is considered as good or bad quality are also different between users. This dynamic picture of user preferences and profiles are considered important to analyze and structure, in order to use part of it as input to optimization mechanisms.

In addition to the specific preferences of a user, there are also other differences in terms of factors contributing to the per-user QoE. Users are, e.g., different in terms of expectations concerning real quality. This may be directly related to the preferred user terminal capabilities or just basic differences in human perception. User preferences are also influenced by cost factors and assumed user rank in the specific home environment.

3.2 Service Objects

Many Internet-based services have certain requirements in terms of what is needed in order to be used. These requirements have traditionally been described by QoS parameters (delay, packet loss, jitter and bandwidth). This set of service information is still valid, but should be extended with additional parameters. This is especially important in light of the rapid evolution in content delivery techniques and associated technologies. The concept of adaptive streaming is an example of this. In this scenario the quality levels of a service is able to adjust itself according to end-to-end performance before and during service usage. This makes the bandwidth requirement for a certain service no longer fixed, but rather a variable parameter with some min/max thresholds and granularity. Further on, the concept of multi-source streaming from distributed and shared service platforms is also growing in popularity making it more challenging to recognize and classify services. The distribution of sources also makes the services become less sensitive for high delay, packet loss and jitter as it can pick the best performing streams and compose the service based on this.

3.3 Reasoning Process

In order to see the effect on using end user and content provider knowledge in the optimization process, three different schemes have been studied. These schemes are to some extent in line with the concept of a DiffServ bandwidth broker [13], but instead of priorities and policies as basis for bandwidth sharing we are using other knowledge components.

The first scheme is the basic FCFS (First Come First Served), where all knowledge use has been disabled and the home gateway operates in a regular best effort mode. The second mode is named STOPINC, where the Action Plane prevents background traffic source from increasing (if attempted) during a period where an end user preferred service is running below its maximum level. The third mode is called STEPDOWN, where the Action Plane in addition to what the STOPINC mode does - also makes a background traffic source decrease its rate according to the end user preferred service granularity. In the latter mode, the purpose is then to make it easier for the preferred service to increase its rate – one step closer to its maximum. For both the STOPINC and STEPDOWN modes, when a background traffic source is either prevented from increasing or even made decrease its rate – it will be subject to this control for a certain period. This period should be enough so that the adaptive source notices that there is a chance of increasing rate.

4. Simulation Model

The user scenario modeled in the simulator is a residential user group present in a typical home environment. The user group is connected to the Internet through a typical broadband connection. The broadband connection represents the resource shared between users and associated service.

A group of 4 users are considered, each of which operate independently of each other. Each user can start a single service at a time. There are no feedback mechanisms implemented in terms of users changing behavior as a result of good or poor performance.

Table 2. Simulation parameters

Parameter	**Adaptive Service**	**Bkgd. Service**
Max sessions	1	3
Bitrate (Kbps)	300-900	100-2800
Granularity (Kbps)	300	1
Time to first start (s)	Uniform(3,10)	
Session lifetime (s)	Uniform(10,30)	
Session lifetime (s)	Uniform(3,10)	
Conn. capacity (Kbps)	1000-7000	
Control Period (s)	3	
Simulation time per run	7 days	
Number of seeds / run	100	

As can be seen from the simulation parameters, the session lifetimes are very short – much shorter than what could be expected in real life. The purpose of this was to make the simulation scenarios as dynamic as possible.

In order to see the effect of the studied QoE optimization process during different levels of congestion, the access capacity was varied while the service characteristics are kept the same.

The simulator was built using the process oriented Simula [14] programming language and the Discrete Event Modeling On Simula (DEMOS) context class [4].

4.1 Adaptive Service

The adaptive service (cf. Figure 3) operates between the max/min thresholds of respectively 900Kbps and 300Kbps and the granularity of increase/decrease could be set to values between 50Kbps and 300Kbps in the simulator - corresponding to fine versus course rate granularity. The granularity used in the presented results is 300Kbps and the interval of potential rate change was set to 2 sec. The reason for choosing both these values was that these are common parameter values used by live services [2][16].

The adaptive service will always try to increase its rate if possible, and will remain at max level when reached until it finishes unless if influenced by background traffic bursts. The influence from traffic bursts has been included in the model as it would be difficult to prevent, due to the location of the optimization process in the home gateway after the downstream congestion point.

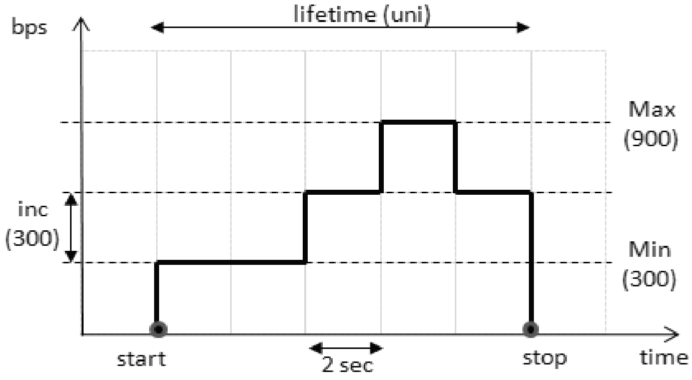


Figure 3. Adaptive service profile

The lifetime of the adaptive service session is taken from a random uniform distribution. A single adaptive service is run at a time, with repeated starts/stops during the simulation period.

4.2 Background Service

The background services (cf. Figure 4) used in the simulation operates in a rather simple mode, but potentially close to a worst case scenario. The sources are very bursty and pick a new target rate for each interval between a lower (100kbps) and upper threshold (2800Kbps) according to a uniform distribution. The intervals between each rate change is according to a negexp distribution (λ=1).

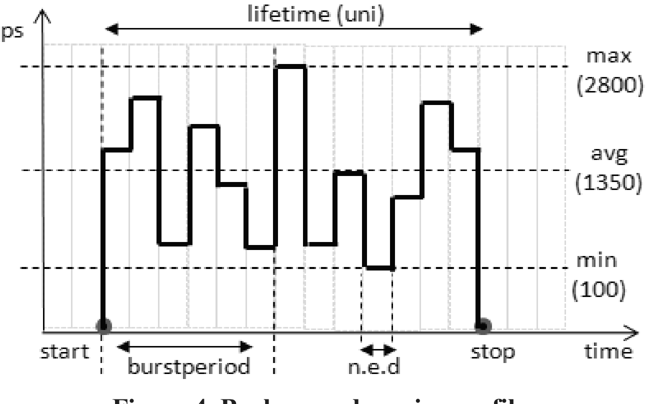


Figure 4. Background service profile

Whenever a background service starts up, it enters a burst period. During this burst period, the background services are allowed to influence the user preferred services, and in the case of congestion – they will make the adaptive service decrease its rate. The reasoning behind this is that the optimization process simulated is placed in the customer home gateway, and therefore after the access congestion point for traffic to the customer.

The duration of the burst period is decided by how fast new background traffic can be put under control by action plane components in the home gateway. Depending on the traffic type (TCP, UDP) and associated application this period will have different values. In the simulation results presented in this paper, the burst period has been varied between zero and 0.6 sec. The value of zero would represent no burst impact (ideal situation).

The lifetime of the background service session is taken from a random uniform distribution. Maximum three background services are run at a time, with repeated starts/stops during the simulation period.

5. Simulation Results

The parameter studied in the simulations is the average achieved bitrate for the adaptive service as a function of access capacity. Traffic load is kept constant.

5.1 FCFS, STOPINC and STEPDOWN results

In Figure 5 results are presented where the burstperiod is set to 0.2 seconds, the adaptive service has increments of 300Kbps and the background services have n.e.d rate increment intervals with λ=1. The three different models FCFS, STOPINC and STEPDOWN are then compared.

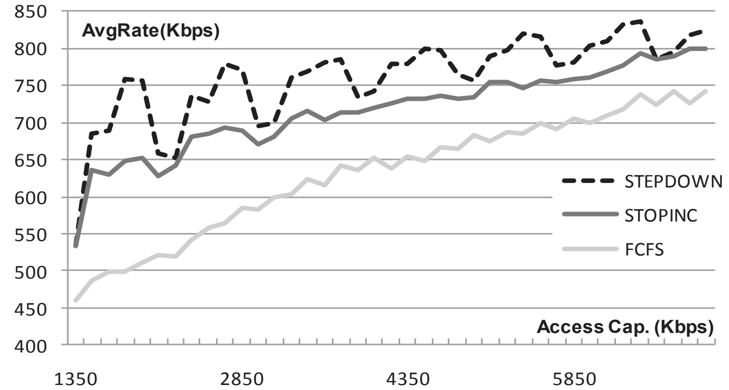


Figure 5. Comparison of optimization models

The 95% confidence intervals for the STEPDOWN model are shown in Figure 6, in order to give see how similar the results from the different simulation runs are.

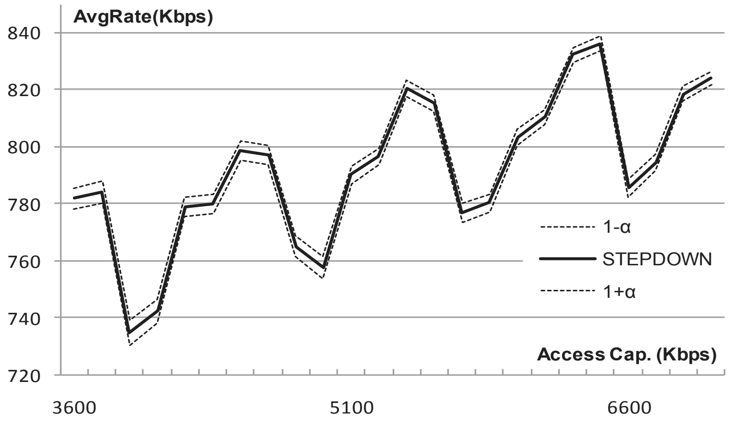


Figure 6. Confidence intervals for STEPDOWN

The confidence intervals are all in the region of +/- 2 to 7 across the studied access capacity range, which is very close to the plotted averages.

5.2 Effect of changing burstperiod

In Figure 7 the effect of changing the burstperiod for the background service is shown for the STEPDOWN optimization model. The purpose of changing this parameter was to see if it had a major impact on the simulation results, and also to provide an indication on how fast the relevant Action Plane components would have to be in order to support the proposed QoE optimization process.

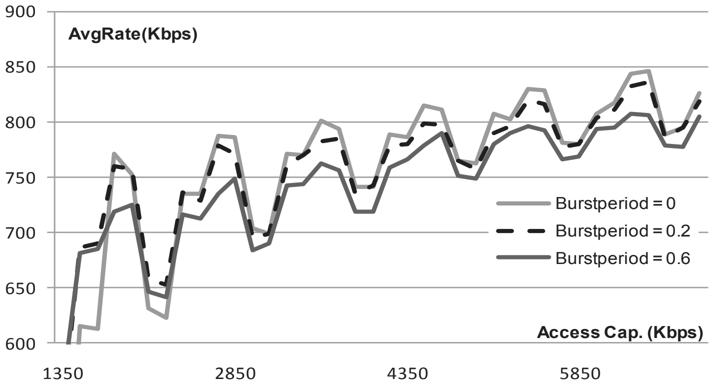


Figure 7. Change of burstperiod for STEPDOWN

The burstperiod values used are 0, 0.2 and 0.6 – whereas the value of 0 corresponds to an ideal scenario where the background services never influences the preferred adaptive service. The higher burstperiod values corresponds to scenarios where the Action Plane require some time interval in order to achieve control on the background services.

6. Analyzing the results

The results presented in the previous section are considered promising, as they support the statement subject to investigation. The comparison between the FCFS, STOPINC and STEPDOWN modes of operation (cf. Figure 5) shows that for a home gateway the potential improvement in average bitrate for a preferred service is significant, if knowledge about the service granularity is made available. The simulation results show that during high congestion both the STOPINC and STEPDOWN models perform significantly better that FCFS. For the STOPINC mode there is a potential for between 10-30% higher average rate, and for the STEPDOWN mode the same potential is between 10-40%. The STEPDOWN mode performs significantly better that STOPINC for all access capacity levels (cf. Figure 8).

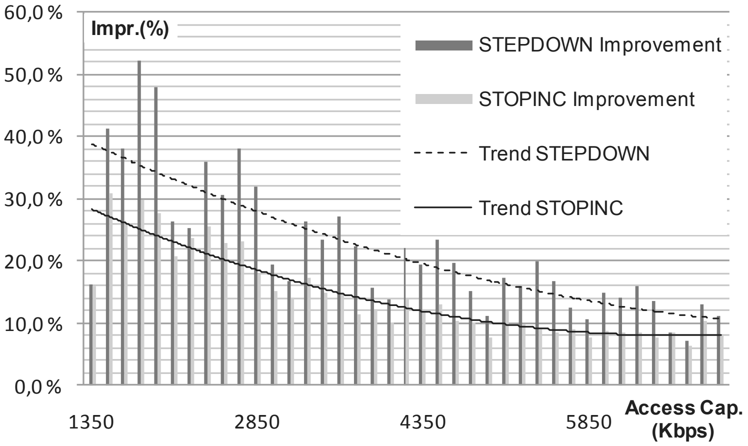


Figure 8. Rate improvements in percentage per model

The results when changing the burst period (cf. Figure 7) illustrate the importance of having an efficient Action Plane supporting the reasoning process. If services are not put under control as fast as possible, it reduces the potential of STEPDOWN in the order of tens of Kbps.

It should be noted that there is no general 1:1 mapping between an achieved value of a QoS metric such as bitrate and a specific QoE metrics. However, it is a fair assumption that there is a weighted mapping between QoS metrics and related QoE metrics, following the preference and perception levels of a certain user. In line with this, we can say that the achieved increase in bitrate for the preferred adaptive service contributes to an increased QoE level.

7. Conclusion and future work

Based on the analysis and simulation results presented, the statement of a potential gain in implementing QoE optimization mechanisms in the access and home network domain is strengthened. It is clear that even with just very basic knowledge components available from the user and service objects (cf. Figure 2) a significant improvement in QoE can be achieved.

The presented results may also have value for pure network oriented QoS mechanisms, if this type of stepwise service adaption becomes a success in emerging service delivery architectures. As an example, it is likely that the bandwidth broker concept of DiffServ could benefit from introducing this type of service knowledge in its operation.

As future work in this area, the plan is to investigate more complex user and service scenarios. It is also the intention to make the service models used in the simulator closer to real life traffic. Further on, the logics in the reasoning process together with efficient action plane components will be addressed.

8. Acknowledgements

The reported work is done as part of the PhD studies for the first author, which is an integrated part of the Road to media-aware user-Dependant self-aDaptive NETWORKS - R2D2 project. This project is funded by The Research Council of Norway

References

[1] A. Aggarwal, S. Savage, and T. Anderson. “Understanding the performance of tcp pacing”. In Proc. IEEE Nineteenth Annual Joint Conf. of the IEEE Computer and Communications Societies INFOCOM 2000, volume 3, pages 1157–1165.

[2] S. Akhshabi, A. C. Begen, and C. Dovrolis. “An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http”. In ACM Multimedia Systems (MMSys), 2011.

[3] E. Areizaga, L. Perez, C. Verikoukis, N. Zorba, E. Jacob, and P. Odling. “A road to media-aware user-dependent self-adaptive networks”. In Proc. IEEE Int. Symp. BMSB ’09, pages 1–6, 2009.

[4] G. Birtwistle. Demos - A system for Discrete Event Modelling on Simula. School of Computer Science, University of Sheffeld,, July 1997.

[5] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski. “A knowledge plane for the internet”. SIGCOMM’03, August 2003.

[6] H. Dequeker, T. V. Caenegem, K. Struyve, E. Gilon, B. D. Vleeschauwer, P. Simoens, and W. V. de Meerssche. Advanced features for mm enabled access platform (muse project deliverable). Technical report, December 2006.

[7] H. Jamjoom and K. G. Shin. “Persistent dropping: an efficient control of traffic aggregates”. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM ’03, pages 287–298, New York, NY, USA, 2003. ACM.

[8] J. O. Kephart and D. M. Chess. “The vision of autonomic computing”. Computer, 36(1):41–50, 2003.

[9] S. Latr´e, P. Simoens, B. De Vleeschauwer, and V. de Meerssche. “An autonomic architecture for optimizing qoe in multimedia access networks”. Computer Networks 53, 10:1587–1602, 2009.

[10] S. Latre, P. Simoens, B. De Vleeschauwer, W. Van de Meerssche, F. De Turck, B. Dhoedt, P. Demeester, S. Van Den Berghe, and E. G. de Lumley. ”Design for a generic knowledge base for autonomic qoe optimization in multimedia access networks”. In Proc. IEEE Network Operations and Management Symp. Workshops NOMS Workshops 2008, pages 335–342, 2008.

[11] S. Latr´e, S. Verstichel, and B. D. Vleeschauwer. “On the design of an architecture for partitioned knowledge management in autonomic multimedia access and aggregation networks”. MACE, pages 105–110, 2009.

[12] K. Nichols, S. Blake, F. Baker, and D. Black. “Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers”. RFC 2474 (Proposed Standard), Dec. 1998. Updated by RFCs 3168, 3260.

[13] K. Nichols, V. Jacobson, and L. Zhang. “A two-bit differentiated services architecture for the internet”. RFC 2638 (Informational), July 1999.

[14] R. J. Pooley. An Introduction to Programming in Simula. Blackwell Scientific Publications. ISBN: 0632014229, 1987.

[15] P. Simoens, B. D. Vleeschauwer, W. V. de Meerssche, F. D. Turck, B. Dhoedta, and P. Demeester. “RTP connection monitoring for enabling autonomic access network qos management”. In Proceedings of the 12th European Conference on Networks and Optical Communications (NOC), 2007.

[16] TV2. Adaptive webtv streaming solution from TV2 Norway. Available online at www.tv2sumo.no. Last access 01.01.11.

[17] B. J. Villa and P. E. Heegaard. “Monitoring and control of qoe in media streams using the click software router”. In Norsk Informatikk Konferanse (NIK2010), Norway. ISBN 978-82-519-2702-4., volume 1, pages 24–33, Nov 2010.

[18] B. D. Vleeschauwer, W. V. de Meerssche, P. Simoens, F. D. Turck, B. Dhoedta, and P. Demeester. “Enabling autonomic access network qoe management through tcp monitoring”. In Proceedings of the First IEEE Workshop on Autonomic Communications and Management (ACNM), 2007.

[19] B. D. Vleeschauwer, P. Simoens, W. V. de Meerssche, S. Latré, F. D. Turck, and S. V. den Bergheb. “Autonomic qoe optimization on the access node”. In Proceedings of the Broadband Europe, 2007.

[20] H.-Y. Wei, S.-C. Tsao, and Y.-D. Lin. “Assessing and improving tcp rate shaping over edge gateways”. IEEE Trans on Computers, 53(3):259–275, 2004.

PAPER 3: A Monitor Plane Component for Adaptive Video Streaming

Bjørn J. Villa, Poul E. Heegaard

Published in
Proceedings of Norsk Informatikkonferanse 2011 (NIK 2011)
Tapir Akademiske Forlag, ISBN 978-82-519-2843-4. pp. 145-154
Nov. 21-23, 2011, Tromsø, Norway

A Monitor Plane Component for Adaptive Video Streaming

Bjørn J. Villa, Poul E. Heegaard

Abstract: Video components have become an important part of many popular Internet services, and at an aggregated level they are now a dominant service type in terms of generated traffic volume. Many of these services are provided from content providers utilizing the best effort part of an Internet service. In this paper we provide an introduction to the technical evolutions in this area and also present the capabilities of the more recent enhancements in terms of adaptive streaming services. These services are put into the context of an established QoE optimization framework, and a candidate Monitor Plane (measurement and reporting) component is described. The potential use of this component in order to provide an objective QoE indicator and as input to a home gateway bandwidth broker is proposed.

1. Introduction

Video streaming over the Internet has experienced a tremendous growth the past decade. As an example of such a service YouTube may be considered. This service was established in February 2005. By early 2007 - YouTube alone represented about 20% of all HTTP traffic, or nearly 10% of all traffic on the Internet [4]. A recent forecast report [17] states that by 2012 more than 50% of all the traffic will be video based.

As business models for premium content delivery Over-The-Top (best-effort Internet based) are appearing, methods for assuring the quality levels even in this domain are becoming more important. There is a shift from focusing only on the Quality of Service (QoS) dimension to also investigate the Quality of Experience (QoE) dimension [2]. The latter would represent the actual user experience and is therefore considered as more suited for measuring the overall user satisfaction with a service.

A concept which has been introduced for streaming services with promising results is the use of adaptive streaming [21]. This type of streaming allows the client to dynamically change the video quality of an ongoing stream. This is done in order to adapt to changes in network conditions. However, it has been shown [1] that even this solution has its shortcomings as it does not address the situation where multiple service instances are competing. One way of addressing this problem is to seek the assistance of an external entity which is shared by the competing clients. In a home network environment, this would be the home gateway. In order to be able to do such operations, the home gateway must have a way of obtaining information about active video streams.

The structure of this paper is as follows. Section 2 introduces a framework for QoE optimization; Section 3 describes video streaming technologies; Section 4 describes methods for information collection; Section 5 presents a Monitor Plane component for the MS SilverLight framework; Section 6 presents the experimental implementation; Section 7 presents the potential use of the Monitor Plane reports; Section 8 presents the conclusion and Section 9 outlines future work.

2. QoE optimization framework

There have been several proposals for autonomic access networks being able to perform optimization tasks on both network and service level [6] [10] [19] [20]. These solutions mainly consist of three types of layers or components each serving different roles. These components are called Knowledge Plane (KP), Monitor Plane (MP) and Action Plane (AP).

The idea is that network components through information gathering in MP should be able to recognize, interpret and classify current traffic patterns. The reasoning process is done by the KP which allows the network to identify potential problems in a service mix. The KP should then be able to use a diverse set of tools to resolve the problem it has identified, and perform an optimization towards a specific objective. The actions chosen by the KP is realized by the AP which applies a change in the network configuration or make some adjustment to selected services [20]. In Figure 1 the placement of KP, AP and MP components re illustrated.

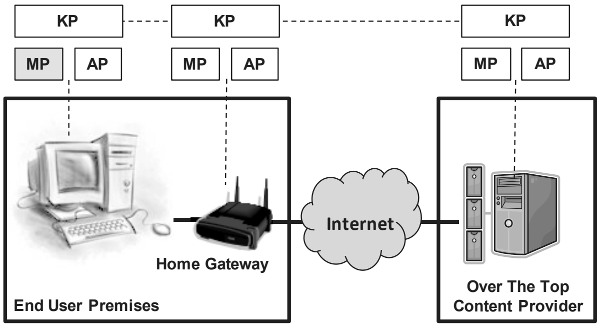


Figure 1 QoE Optimization Framework

The focus in this paper is to describe the Monitor Plane component located in the end user client, and how this can be implemented for emerging video streaming services.

3. Video streaming solutions and technologies

A video streaming service gives a client the opportunity to access video and audio content over the Internet. The streaming content is made available by service providers with the use of streaming servers. The distributed content is either real-time (scheduled) or on-demand. The challenge of providing these services in a best effort network environment is to handle the frequent fluctuations in network characteristics such as available bandwidth and experienced delay/jitter. This section provides an overview of technical approaches and concepts used for video streaming, both historically and today.

3.1 Video streaming solutions and technologies

The Real-Time Transport Protocol [15] provides end-to-end network transport functions for real-time multimedia content (audio, video, etc.). RTP is only a transport protocol and has no means of providing interactivity to the client. In order to achieve such interactivity – giving the users an opportunity to play, pause and fast forward, RTSP [16] has to be used. Hence, RTSP can be considered an extension of the RTP protocol.

Both RTP and RTSP can use RTCP [15] for probing the quality of active sessions. RTCP is based on periodic transmission of control packets. The feedbacks which can be gathered from using RTCP are number of lost packets, jitter and RTT. These parameters can then be used by RTP to perform adaptations to the quality or transmission rate to be used for the data distribution [10].

The reason for RTP/RTSP not being as popular as expected is that it runs on special ports which could be blocked by firewalls and NATs. Another disadvantage is that it requires special servers configured for this technology. In order to overcome these issues, the concept of HTTP based streaming has emerged.

3.2 TCP in media streaming

Earlier it was believed that TCP did not have the properties needed in order to provide media streaming because of the contribution to extra delay when retransmission is activated. Another reason for the concern about the streaming capabilities of TCP was the congestion avoidance algorithms used. It was believed that congestion avoidance would contribute to large fluctuations in the video bit rate and viewers QoS/QoE.

However, it has been shown that TCP can be used for video streaming – and it is actually quite commonly used for this purpose. The two main methods of dealing with the retransmission issue are buffering and bit rate adjustment. Offering the video in a set of different bit rates will allow the client to choose a bit rate best suited to its traffic condition. This way the client can handle large bit rate fluctuations, and in combination with a buffer also deal with retransmission delay. Using custom players we can obtain smooth transitions between bit rates. The idea then is that having a smooth transition to a lower bit rate gives a higher QoE than if the video was to stop playing in order to fill the buffer again. These properties in combination with TCP functions such as fairness are why TCP is being used as the transport protocol of choice for video streaming [9].

3.3 Progressive download

There are several solutions to provide media streaming. One is to transfer the video content from the streaming server to the client using a TCP session at the highest possible bit rate. A buffer is then used at the client-side which starts playing the video content after some byte threshold in the buffer is reached [3]. This will reduce the clients waiting time, compared to downloading the full length video file, but it will not perform well under fluctuating network conditions. This can be prevented using large buffers, but the client waiting time until it starts playing the video content will approach the video download time, which again reduce interactivity. This technique is referred to as progressive download and is used by services such as YouTube.

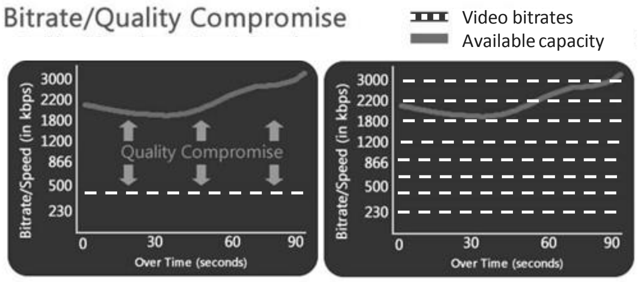
3.4 Single rate adaption

An enhancement made in traditional streaming was to determine the user bandwidth during startup of the streaming session. This allows for choosing a custom video quality for a specific user. Using the bandwidth measurements, a static bit rate is chosen for the client. Some fluctuations in the network conditions can be handled by using this technology in combination with a client side buffer. This approach performs well under rather stable network conditions, but it does not cope well with large fluctuations in network conditions. This technique can be referred to as single rate adaption [8]. A further enhancement of this is the concept of self-adaptive streaming which will be presented in the following section.

3.5 Self adaptive streaming

Self-adaptive streaming is a concept developed for handling fluctuations in a network environment. In general the streaming server advertises a set of available streams with distinct bit rates and other relevant parameters to the client, using a manifest file written in a markup language like XML. The streaming server achieves different bit rates by dividing a video file into small segments (some seconds long) with different resolution, hence file size. The client will then monitor network parameters such as display resolution, CPU usage, available bandwidth, etc. Based on these parameters the client will use HTTP GET requests to retrieve segments with the most appropriate bit rate in accordance to the current network conditions [21].

The network conditions are monitored periodically to detect fluctuations and the switching between different bit rates is done seamlessly (cf. Figure 2). It is believed that this is the technique of choice as it does not require any changes to the current Internet architecture. Another advantage is that by using adaptive streaming there is a possibility to constrain the buffer size and therefore reduce the wasted bandwidth if a client decides to stop watching the video half way in.



Traditional Streaming
(Progressive / Single Rate Adaption)

Self Adaptive Streaming

Figure 2 Traditional vs. Adaptive Streaming

This technology can adapt to different users with different connections to the Internet. All the users can view the media without the need for the media itself being scaled for the user with the lowest bandwidth. Hence, a client with high bandwidth can view a video with high quality, while clients with low bandwidth can view videos with lower quality

A negative property of adaptive streaming is that there are no standards for this technology, but there is ongoing standardization work. This has resulted in an IETF draft [13]. Some examples of proprietary solutions using this HTTP streaming are Adobe Dynamic HTTP Streaming, Apple HTTP live streaming and Microsoft Smooth Streaming (SilverLight framework).

3.6 HTML5 video streaming

HTML5 is not a web standard yet, but still it is of great interest to the future of video on the web [18]. Most browsers support the video tag introduced in HTML5 which allows for easy addition of video in a web site, without the need for browser-specific third-party plug-ins. The problem with the plugins is that the video players and content were a black box, as seen from the browser. This gives some challenges in terms of applying style sheets or transitions to the video being showed on the web page [5]. In less technical terms, the integration of the video component and the rest of the web page is not optimal.

The question then becomes if HTML5 will be able to provide the wanted functionality that the proprietary plug-ins already do. The usability across platforms is a plus, but to replace the proprietary technologies it must also provide technology that makes it a viable substitute.

4. Methods for information collection

In this section we will describe how a Monitor Plane component for streaming services can be implemented in a home network domain. There are several approaches to this, including both active and passive methods. The location of these can also vary, whereas both end systems and intermediate nodes are possible locations.

4.1 Monitoring traffic

The traffic monitoring approach contains both a traffic capture and an interpretation part. The goal is to make use of the information that is transmitted between the client and server, by an intermediate node. In adaptive streaming services there is usually an initial message sequence where the server advertises its services and bandwidth options. In addition, the client often provides information for authentication. As mentioned in section Y, the former can be done using a manifest file. The latter is often done in a similar matter with XML encoded strings presenting the user information to the server. As long as this information is transmitted as clear text with a well-known format, an intermediate node such as the home gateway could capture this information and use it for the purpose of QoS/QoE optimization.

4.2 Altering streaming client code

A tempting alternative to the traffic monitoring approach is to collect all required information directly from the client. In this scenario one would be less vulnerable for changes in message format, and also the clear text requirement would then disappear. The reason for this is that the video streaming clients are downloaded from the service provider every time the user watches a video. This means that users would get the software and successive upgrades without noticing it.

4.3 Information from the web browser

One thing the clients have in common is that they are all viewed in a web browser, and most web browsers allow for custom third-party plug-ins or add-ons. By creating an add-on for information collection and reporting purposes we could retrieve information about the resource consumption and serviced running. This method has several imperfections which in our view could disqualify it.

First of all, the number of browsers is quite large. A custom add-on would have to be implemented for each browser and constantly be updated as the browsers often have version upgrades. In addition, video player plugins based on e.g Silverlight or Flash are black boxes to the browser – and therefore information collection from these objects are not possible.

4.4 Reporting application

Another solution for the active approach is to have a standalone application running on the end system. This application could be written using any programming framework and language. This application would run in the OS directly and therefore give easier access to a lot of OS parameters such as CPU load and available memory. In addition we could get information on what type of interface the end system uses to connect to the home gateway, which is interesting since e.g. wired and wireless Internet connections have different levels of QoS. The network link load could also be collected.

The standalone approach has some advantages in retrieving network related and OS specific information. However, the retrieval of application information would be a significant challenge. Application specific information, like the current quality level used by a Smooth Streaming application, will be difficult to retrieve. The application is run in a black box, on top of a browser. It is not made for access by other applications.

5. Monitor plane component for adaptive streaming

The self-adaptive streaming solution chosen for closer study was based on the Microsoft SilverLight framework. The reason for this choice was the use of it by a research project partner (TV2) in their commercial product. A Silverlight based web page allows code to be added for more than stream delivery. For our purpose we added code which observed parameters of the video player and some platform aspects (cf. Table 1). This approach is in line with the concept described in section 4.2. Whenever changes in these parameters were observed – or at regular intervals, a report would be sent.

Parameter	**Description**
event	The type of event triggered in Silverlight client
eventMessage	A message in relation to the event
playState	Current playstate of the Silverlight player
fullScreen	Boolean value indicating if player is full screen
clientId	A Silverlight client globally unique identifier
avgProcessorLoad	Processor load of the client computer
avgProcessLoad	Processor load caused by Silverlight
videoDownloadBitrate	Current rate of streaming
avaliableVideoBitrates	Video bit rates the media is offered in
timeIncrement	Time since application was loaded, in seconds
sourceIpAndPort	IP and port of streaming server, as DNS name

Table 1 Information components in SilverLight video player

In addition to the information available directly from the SilverLight video player, a set of additional parameters can be collected by some additional processing (cf. Table 2). This would typically be a software process on the server side which inspects the incoming HTTP POST messages, extracts the source IP address and perform a lookup toward a whois service in order to obtain the AS number.

Parameter	Description
myIP	IP address of the streaming client
ispAS	The AS number for the client IP, identifying ISP

Table 2 Additional client information components

Further on, the Silverlight code uses a set of event handlers to notify if changes have occurred to the player or video stream. When such an event occurs, methods are invoked to retrieve a set of system parameters. These parameters will be retrieved and sent when the video player triggers a PlayStateChange, FullScreenChange, BitRateChange, Initial- or PeriodicEvent. This allows us to monitor what it is currently running and also be notified when it changes.

The combined information from the Silverlight event and system parameters can then be passed as HTTP POST message to an external entity. The external entities of special interest in the context of QoE optimization is the home gateway, and for analysis - a data collection / statistics server.

6. Experimental implementation

In order to test the proposed concept of having a Monitor Plane component integrated with an adaptive streaming service a prototype implementation has been developed [7]. This prototype has been implemented in a lab environment (cf. Figure 3). The lab setup is supposed to be similar to a typical home environment, with a number of PC's – a home gateway (experimental router) and a broadband access with a limited capacity set by the access bandwidth control node.

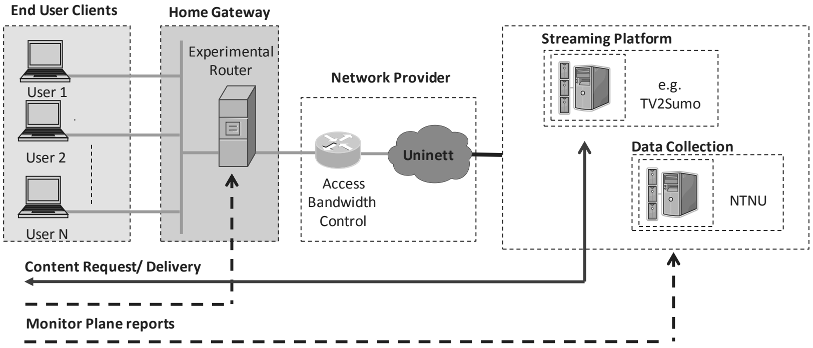
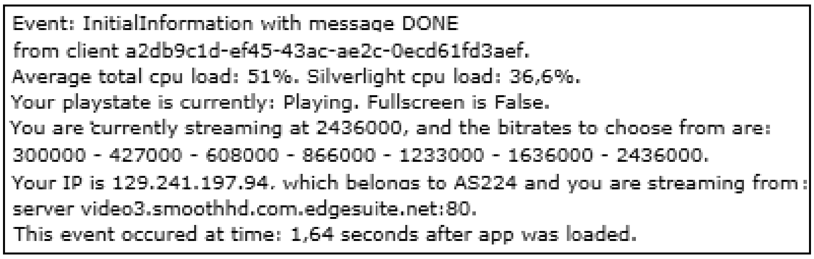


Figure 3 Monitor Plane component testbed

The prototype Monitor Plane Component in the adaptive streaming application used in the testbed is integrated with live TV channels from TV2 Norway, using their content delivery solution for webTV called TV2Sumo. In the testbed the Monitor Plane messages are sent both to an experimental home gateway and to a central statistics server. This has given a good basis for evaluating the feasibility of having such a Monitor Plane component present.



```
Event: InitialInformation with message DONE
from client a2db9c1d-ef45-43ac-ae2c-0ecd61fd3aef.
Average total cpu load: 51%. Silverlight cpu load: 36,6%.
Your playstate is currently: Playing. Fullscreen is False.
You are currently streaming at 2436000, and the bitrates to choose from are:
300000 - 427000 - 608000 - 866000 - 1233000 - 1636000 - 2436000.
Your IP is 129.241.197.94. which belongs to AS224 and you are streaming from :
server video3.smoothhd.com.edgesuite.net:80.
This event occured at time: 1,64 seconds after app was loaded.
```

Figure 4 Monitor Plane report content

A typical Monitor Plane report is illustrated in Figure 4. The use of these reports in respectively the home gateway and statistics server will be described in the following section.

7. Using the monitor plane reports

There are several areas where the Monitor Plane reports could be used, but in the context of this research the focus is on how to optimize QoE. In order to perform such optimizations it is necessary to have methods for both analysis and control. The analysis part of objective QoE in our scenario is done in the data collection / statistics server, while the control part is foreseen done in the home gateway by means of a bandwidth broker [12] function.

7.1 Input to objective QoE metric

There is a wide range of metrics which influence how satisfied an end user is with a service such as e.g. video streaming. One such QoE metric which is closely related to network aspects is end user perceived fairness. Research from the social science and psychology domain [11] states that perceived fairness is closely related to what is called ‘Social Justice’. In this context a queuing system or any other resource allocation mechanism would be considered as a ‘Social System’. It has further been found that users react negatively to any system behavior which gives better service to other user, unless justification is provided.

The end user notion of system discrimination has been suggested by [14] as an important measure of perceived service quality, and more specifically the perceived fairness is stated to be closely related to the discrimination frequency. Applying the concept of discrimination to adaptive streaming, it would be related to situations where end user expectations are not met over time, but also to any negative change in the service quality (e.g. rate level reductions). Using the Monitor Plane component described in this paper it would be possible for the data collection / statistics server to process the incoming reports per stream, and present a perceived fairness metric based on e.g. the number of rate level reductions per minute. This metric could be considered as an indicator of objective QoE for the service.

7.2 Input to bandwidth broker

Traditional QoS implementations in network elements are based on the classification and differentiation of each IP packet based on relationship to certain priority classes. The concept of a bandwidth broker [12] located in the home gateway could provide significant enhancements in this area. A bandwidth broker could utilize more information about each flow of IP packets and perform bandwidth allocations according to this. The additional information about the available quality levels for adaptive streams would be very valuable for this type of function. In the case where congestion occurs, the home gateway should enforce bandwidth limitations to some of the traffic in order to avoid overall service degradation. Knowing what the levels are for the adaptive services has the potential of making the function more efficient.

The Monitor Plane reports described in this paper could be used as input to this type of function located in the home gateway. The potential gain of doing so has been reported in [20] where simulations have been used to study the effect.

8. Conclusions

The reported work should be considered as one of many potential components in a framework for QoS/QoE optimization. The value of Monitor Plane components as decribed is determined of how well they are applied. The suggested use as input to an objective QoE metric and a bandwidth broker located in the home gateway are those examples which relates most to the specific research area.

The conceptual mode of operation – where end user applications participate in the process of optimization by knowledge sharing is considered as very promising and represents a new approach. In general, it is considered to have great potential of simplifying flow identification and classification, something which in the traditional approach has been very challenging in a dynamic environment. The specific use together with the home gateway represents a novel enhancement to existing Internet based service delivery architectures. The proposal of using perceived fairness as an indicator of objective QoE is an important step in the direction of being able to measure and influence the end user experience. The understanding and definitions of objective QoE is challenging, but is expected to gain increased attention.

9. Future work

As future work it is planned to both enhance the capabilities of the Monitor Plane component, and also the way this information is used by different entities. As the QoE optimization framework consists of many components, it is also required to find a way to combine and reason the information obtained from different sources.

The ability to post-process Monitor Plane reports from a high number of live users are considered as very interesting in order to both understand user behavior and also to study the real performance of this service type.

Investigating the correlations between achieved quality levels and changes in quality levels (perceived fairness) up against completed service delivery could give more important knowledge about objective QoE

10. Acknowledgements

The reported work is done as part of the PhD studies for the first author which is an integrated part of the Road to media-aware user-Dependant self-aDaptive NETWORKS - R2D2 project. This project is funded by The Research Council of Norway.

References

[1] Saamer Akhshabi, Ali C. Begen, and Constantine Dovrolis. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. In ACM Multimedia Systems (MMSys), 2011.

[2] E. Areizaga, L. Perez, C. Verikoukis, N. Zorba, E. Jacob, and P. Odling. A road to media-aware user-dependent self-adaptive networks. In Proc. IEEE Int. Symp. BMSB '09, pages 1–6, 2009.

[3] A. Begen, T. Akgul, and M. Baugher. Watching video over the web: Part 1: Streaming protocols. Internet Computing, IEEE, 15(2):54 –63, April 2011.

[4] Xu Cheng, , Jiangchuan Liu, and Cameron Dale. Understanding the characteristics of internet short video sharing: Youtube as a case study. IEEE TRANSACTIONS ON MULTIMEDIA, 2009.

[5] François Daoust, Philipp Hoschka, C. Patrikakis, Rui Cruz, Mário Serafim Nunes, and David Osborne. Towards video on the web with html5. Oct. 2010.

[6] Hans Dequeker, Tom Van Caenegem, Kris Struyve, Edith Gilon, Bart De Vleeschauwer, Pieter Simoens, and Wim Van de Meerssche. Advanced features for mm enabled access platform (muse project deliverable). Technical report, Alcatel Bell and IBBT - IBCN, December 2006.

[7] Kristian Haugene and Alexander Jacobsen. Network based qoe optimization for "over the top" services. Master's thesis, NTNU, 2011.

[8] Charles Krasic. A framework for quality-adaptive media streaming: encode once - stream anywhere. PhD thesis, 2004. AAI3119036.

[9] Charles Krasic, Kang Li, and Jonathan Walpole. The case for streaming multimedia with tcp. In In 8th International Workshop on Interactive Distributed Multimedia Systems (iDMS 2001, pages 213–218, 2001.

[10] S Latr´e, P Simoens, B De Vleeschauwer, and Van de Meerssche. An autonomic architecture for optimizing qoe in multimedia access networks. Computer Networks 53, 10:1587–1602, 2009.

[11] Hanoch Levy, Benjamin Avi-Itzhak, and Davidz Raz. Network performance engineering. chapter Principles of fairness quantification in queueing systems, pages 284–300. Springer-Verlag, Berlin, Heidelberg, 2011.

[12] K. Nichols, V. Jacobson, and L. Zhang. A two-bit differentiated services architecture for the internet. RFC 2638 (Informational), July 1999.

[13] May Pantos. "http live streaming". ietf draft november 2010. visited: 16.02.2011 <http://tools.ietf.org/html/draft-pantos-http-live-streaming-05>.

[14] Werner Sandmann. Quantitative fairness for assessing perceived service quality in queues. Journal Operational Research, pages 1–34, April 2011.

[15] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003. Updated by RFCs 5506, 5761, 6051, 6222.

[16] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). RFC 2326 (Proposed Standard), April 1998.

[17] Cisco Systems. Cisco visual networking index: Forecast and methodology, 2010-2015. Technical report, June 2011.

[18] Steven J. Vaughan-Nichols. Will html 5 restandardize the web? IEEE Computer, 43:13–15, 2010.

[19] Bjørn J. Villa and Poul E. Heegaard. Monitoring and control of qoe in media streams using the click software router. In Norsk Informatikk Konferanse (NIK2010), Norway. ISBN 978-82-519-2702-4., pp. 24–33, November 2010.

[20] Bjørn J. Villa and Poul E. Heegaard. Towards knowledge-driven qoe optimization in home gateways. ICNS 2011, May 2011.

[21] Alex Zambelli. Iis smooth streaming technical overview. http://www.microsoft.com/silverlight/whitepapers/, March 2009.

PAPER 4: Improving Perceived Fairness and QoE for Adaptive Video Streams

Bjørn J. Villa, Poul E. Heegaard

Published in

Proceedings of The Eighth International Conference on Networking and Services, 2012 (ICNS 2012)

Thinkmind 2012, ISBN 978-1-61208-186-1. pp. 149-158

March 25-30, 2012, St. Maarten, Netherlands

Improving Perceived Fairness and QoE for Adaptive Video Streams

Bjørn J. Villa, Poul E. Heegaard

Abstract: This paper presents an enhancement to a category of Adaptive Video Streaming solutions aimed at improving both Quality of Service (QoS) and Quality of Experience (QoE). The specific solution used as baseline for the work is the Smooth Streaming framework from Microsoft. The presented enhancement relates to the rate adaption scheme used, and suggests applying a stochastic variable for the rate adjustment intervals rather than the fixed approach. The main novelty of the paper is the simultaneous study of both network oriented fairness in the QoS domain and perception based fairness from the QoE domain, when introducing the suggested mechanism. The method used for this study is by means of simulations and numerical optimization. Perception based fairness is suggested as an objective QoE metric which, requires no reference to original content. The results show that the suggested enhancement has great potential in improving QoE, while maintaining QoS.

Keywords: Adaptive Video Streaming; Fairness; QoE

1. Introduction

Solutions for Adaptive Video Streaming are part of the more general concept of ABR (Adaptive Bit Rate) streaming which, covers any content type. The implementation of ABR streaming for video varies between different vendors, and among the more successful one today is the Microsoft Smooth Streaming (SilverLight) framework [1]. In general, the different implementations use many undisclosed and proprietary functions, awaiting results from ongoing standardization.

The basic behavior of adaptive video streaming solutions is that the client continuously performs a measurement and estimation of available resources in order to decide which, quality level to request. The relevant resource from the network side is the available capacity along the path between the server and client. Based on this, at certain intervals the client decides to either go up or down in quality level or remain at the current level. The levels are predefined and communicated to the client by the server at session startup. The changes in quality levels are normally done in an incremental approach, rather than by larger jumps in rate level. The rationale behind this is the objective to provide a smooth watching experience for the user. However, it may also be related to the CPU monitoring done by the client, as this is a key resource required. It may be the case that even if the network can provide you with a much higher rate level, the CPU on the device being used would not be able to process it. During the initial phase of an adaptive streaming session the potential requests of change in rate level are more frequent than later on when operating in a more steady-state phase. To some extent this is a rather aggressive behavior from a single client which, may have undesirable inter-stream impacts. At the same time, in order to give the user a good first impression and make him want to continue using the service it is desirable to reach a high quality as soon as possible.

Among the strongest drivers for commercial use of ABR based services on the Internet are Over-The-Top content providers. These are providers which, rely on the best effort Internet service as transport towards their customers. Therefore, technologies aiming at making services survive almost any network state are of great interest. In addition to focus on the network based QoS dimensions of services and involved networks, there is also a growing interest in the QoE dimension [2]. The latter should be considered as not only a richer definition of quality, but also more focused towards who decides whether something is good or bad, i.e., the end user. The evolution of successful services on Internet indicates that the focus on QoE for Over-The-Top providers is a good strategy

1.1 Problem Statement

The concept of Adaptive Video Streaming is without a doubt very promising. However, as more and more services are adopting this concept the success brings new challenges. The first challenge with effects visible to the end users is how well these services behave when they compete for a shared resource, such as the broadband access to a household. With a strong dominance of video based service on the Internet this issue is important to address. As each client operates independently of each other, it has no understanding of the traffic it competes with. Different clients consider each other as just background traffic. This leads to unpredictable and potentially oscillating behavior of each session, especially in a home environment this type of interference is likely to have a very negative impact on each user QoE.

1.2 Research Approach

The method investigated in this paper to address the problem at hand is to apply specific changes in the algorithm used by each ABR client controlling the adaptive behavior. The specific change suggested is related to the rate adjustment interval used [1]. The effect of changing the duration of the rate adjustment interval from a fixed value T to some stochastic variable is presented and analyzed.

The ABR solution used as reference point for the work is the one from Microsoft (Smooth Streaming). However, the key principles would still apply to other solutions based on similar principles.

1.3 Paper Outline

The structure of this paper is as follows. Section II provides an overview of methodology and metrics; Section III describes the simulation model; Section IV presents simulation results; Section V gives an analysis of the results; Section VI provides the conclusions and an outline of future work.

1.4 Related Work

It has been shown in [3] that competing adaptive streams can cause unpredictable performance for each stream, both in terms of oscillations and ability to achieve fairness in terms of bandwidth sharing. The experimental results presented give clear indication on that competing ABR clients cause degraded and unpredictable performance. Apart from this paper, the topic at hand does not seem to have been addressed by the academic research community to the extent it deserves.

In another paper [4], the authors have investigated how well adaptive streaming performs when being subject to variable available bandwidth in general. Their findings were that the adaptive streams are performing quite well in this type of scenario except for some transient behavior. These findings do not contradict the findings in [3] as the type of background traffic used do not have the adaptive behavior itself, but is rather controlled by the basic TCP mechanisms.

Rate-control algorithms for TCP streaming in general and selected bandwidth estimation algorithms are described in [5]. This work is relevant to any TCP based application delivering a video stream.

In some of our own previous work we have described and analyzed how competing adaptive streams can be controlled using a knowledge based bandwidth broker in the home gateway [6] [7].

2. METHODOLOGY AND METRICS

In this section, we introduce the relevant performance metrics and together with motivation for the chosen focus. Thereafter, some candidate methods on how to improve the performance metrics are given, and finally, the specific method subject for study is presented.

2.1 Flow Based Performance Metrics

For transport flows it is common [8] to focus on the following metrics in order to assess their performance: inter-flow fairness, stability and convergence time. This in addition to the general QoS metrics: bandwidth, packet loss, delay and jitter. The same metrics can be applied to adaptive video streams as they by definition also are flows with similar concerns. The analysis of these metrics can be done from a strict network oriented perspective (QoS), but to some extent also bridged over to a user perception domain (QoE). When focusing on the inter-flow fairness metric this is traditionally analyzed [9] using, e.g., the Jain’s fairness index [10], the product measure [11] or Epsilon-fairness [12] for flows with equal resource requirements. For flows with different resource requirement, the Max-Min fairness [13], proportional fairness [14] or minimum potential delay fairness [15] approaches are commonly seen. Real life adaptive video streams would typically belong to the last category.

Max-Min fairness: The objective of max-min fairness is to maximize the smallest throughput rate among the flows. When this is met, the next-smallest throughput rate must be as large as possible, and so on. Max-min fairness can also be explained by considering it as a progressive filling algorithm, where all flows start at zero and grow at the same pace until the link is full. With this approach the max-min fairness gives priority to the smallest flows. The least demanding flows always have the best chance of getting access to all the resources it needs.

Proportional fairness: The original definition of proportional fairness comes from economic disciplines [14] for the purpose of charging. The original definition is used in the relevant RFC [9] but it does not come across as very constructive for the purpose of analyzing fairness in single resource (e.g., bandwidth) sharing among flows. In this context more recent definitions and interpretations are more suitable [16]. The principle of this would be that a resource allocation is considered proportional fair if it is made to the flow which, has the highest ratio between potential maximum resource consumption and its average resource consumption so far. A further simplification would be to use the current resource usage (if greater than 0) instead of the average in the ratio calculation. The same ratio numbers for each flow could then be used to give a view on the current system fairness by comparing them. If they are all equal the system could be stated as proportionally fair.

Minimum potential delay fairness: The idea behind minimum potential delay fairness is based on the assumption that the involved flows are generated by applications transferring files of certain sizes. A relevant bandwidth sharing objective would be to minimize the time needed to complete those transfers. However, this does not apply to an adaptive streaming scenario and is therefore not discussed any further.

2.2 Perception Based Performance Metrics

There is a wide range of metrics which, influence how satisfied an end user is with a service such as e.g., video streaming. Many of these are not related to network aspects, and therefore difficult to influence by means in this domain. However, one of the perceived performance metrics which, could be correlated with network aspect is the notion of perceived fairness. It is then of great interest to try and find methods of influencing this in a positive manner.

Looking at fairness from an end user perception, research from the social science and psychology domain [17] states that this is closely related to what is called 'Social Justice'. In this context a queuing system or any other resource allocation mechanism would be considered as a 'Social System'. It has further been found that users react negatively to any system behavior which, gives better service to other user, unless justification is provided. Such system behavior is considered un-fair, i.e., in violation with the social justice of the system as the end users considers it as discrimination.

The end user notion of system discrimination has been suggested by [18] as an important measure of perceived service quality, and more specifically the perceived fairness is stated to be closely related to the discrimination frequency. It should be noted that analyzing this type of end user perceived discriminations has a challenge in terms of handling the false positive and false negative cases.

Applying the concept of discrimination to competing adaptive streams, it would be related to situations where end user expectations are not met during steady state periods and also negative changes in service delivery during more transient periods. In other words, whatever makes the end user think that he is being discriminated due to other users in the system, will lead to reduced perceived service quality.

In order to use this type of perceived end user discrimination as a measure for how well the algorithm which, controls the adaptive streams are performing, a clear definition regarding what end users are considering as discrimination is required. This could, e.g., be periods with session rate below some threshold, any change in session rate to a lower level or the session rate change frequency.

2.3 Methods for Improving Performance

There are several things that one could try to incorporate into the adaptive algorithms controlling the ABR service [1] in order to make them perform better in a multi-stream scenario.

The selected performance metrics to be studied are from the network side proportional fairness, and from the end user side the perceived fairness metric as earlier described. Whether it is possible to improve both these fairness metrics at the same time will be an important part of the results.

Randomization of time intervals: The fixed rate adjustment intervals (*T*) used by each adaptive stream while in steady-state may be a contributing factor to inaccurate estimations of available bandwidth and thereby oscillating behavior. An alternative to fixed intervals would be to randomize them by using a per-session stochastic parameter (within certain reasonable bounds). By doing so the available bandwidth estimation methods may become more accurate.

Back-off periods: Whenever a service is reducing its rate level due to observed congestion it may try to increase again after the same amount of time (*T*). In addition to the previous described randomization of this interval, one could also consider introducing a back-off period. This would imply that after a service has reduced its rate level, it enters a back-off period of a certain duration during which, no increase is allowed.

Threshold based behavior: Rather than using the same intervals of potential rate changes all the time, one could introduce a threshold for when it operates more or less aggressive. This threshold could be the mean available rate level for a specific session, or even a smoothed average value for the actual achieved level. This concept is applied with success in more recent TCP versions for the purpose of optimizing performance.

The method chosen for the simulations is according to the first approach described, i.e., a randomization of the intervals between each potential rate change as originally suggested in [3]. As baseline for the simulations, the fixed interval with *T=2s* has been used. Then as stochastic alternatives, both a uniform distribution and a negexp distribution have been implemented. The uniform distribution gives values of *T* between [1.6, 2.4]s, while the negexp alternative gives values of *T* according to the distribution function with λ=0.5 and expected value (1/ λ) = 2s.

3. SIMULATION MODEL

As the adaptive streaming solutions of today are highly proprietary, the details concerning their implementation are not disclosed. Due to this, there will always be some degree of uncertainty concerning their internal functions.

3.1 Assumptions

One of the key functions of an ABR client is the method used for determining whether to go up or down in rate level during times of varying available bandwidth. From studying live traffic it does not seem as if the clients use additional network probing beyond the actual information obtained through download of video segments. Further on, in the likely absence of a per stream traffic shaper at the server side (for scalability and performance reasons), it will give a traffic pattern for each stream which, typically contains a sequence of busy and idle periods. The measured busy period rate is then higher than the actual stream rate level. Also, it is likely that there will be sub-periods within the busy periods where per packet rate is close to the total available bandwidth. As such, the client can probably obtain a rather accurate indication of maximum available bandwidth by just looking at minimum observed inter-arrival time of packets of known size belonging to the same stream.

However, not all streams will have interleaved busy periods so there is a good chance for each stream to overestimate the potential for additional bandwidth. There is a wide range of bandwidth estimation methods and a few of these are described in [19], but again - as the details of the adaptive streaming solutions are not disclosed we will not discuss this part any further. Independent of which, method being used, there will be some degree of uncertainty which, contributes to variable performance. Further on, we assume the following to be true for the ABR sessions to be studied

- No stream coordination at server side
- No involvement from mechanisms in the network between the client and server
- All clients operate independently and do not communicate
- All clients are well behaved in the sense that they follow the same scheme
- At each defined stream rate level there are no variations due to i.e., picture dynamics

3.2 Session Type and Schedule

The ABR sessions used in the simulator are based on profiles observed in commercial services. The quality levels defined are {0, 250, 750, 1500, 2500, 3500, 5000} Kbps. All sessions are of the same type. The sessions are initiated by 10 different users and start time scheduling are done according to stochastic distributed parameters t_a – Uniform [0, 2000] ms and t_b – Uniform [0, 60] s. This gives that all sessions start during the first 60 seconds (t_b), but shifted by some milliseconds (t_a) in order to avoid synchronization of the rate adjustment intervals.

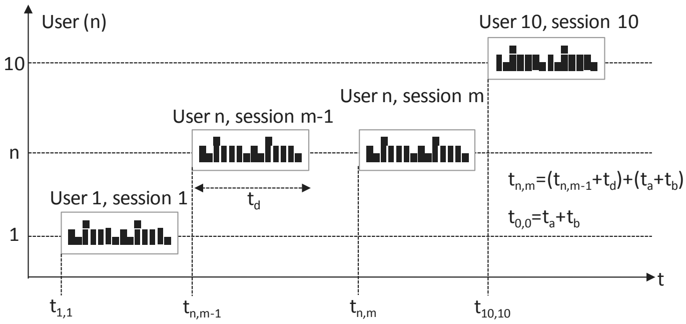


Figure 1. Session scheduling per user

During one simulation run, each user executes a total of 10 sessions sequentially. Time for starting the next session (m) for specific user (n) is noted $t_{n,m}$ (cf. Figure 1). The duration of each session t_d is deterministic and set to 40 minutes. A total of 10 simulation runs using different seeds are executed, corresponding to an aggregated session time of approximately 66 hours per user.

3.3 Rate Adaption Algorithm

The model for rate adaption per session is based on periodic estimation of available bandwidth $A_s(t)$ and calculation of a smoothed average $SA_s(t)$.

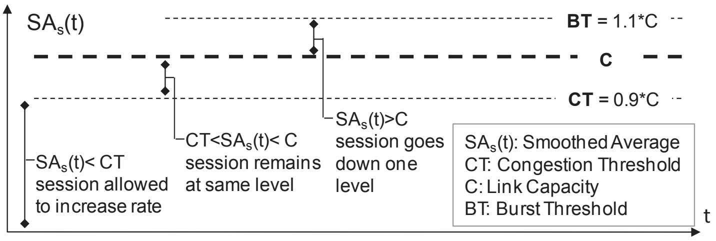


Figure 2. Thresholds for smoothed average

This smoothed average (cf. Figure 2) is compared to a congestion threshold (*CT*), the link capacity (*C*) and a burst threshold (*BT*) in order to trigger a rate adjustment.

Whenever the sum of requested rates from sessions is above the burst threshold (*BT*), the next session which, calculates $SA_s(t)$ will be forced down, independent of the value of $SA_s(t)$. This function is implemented in the simulator in order to incorporate the somewhat unpredictable behavior during times of heavy congestion.

The calculation of smoothed average $SA_s(t)$ is based on [3], and is expressed in (1). The parameter δ gives the weighting of the estimated available bandwidth for the two periods included in the calculation.

$$SA_s(t) = \delta\, A_s(t_{i-1}) + (1-\delta)A_s(t_i) \qquad (1)$$

The available bandwidth estimation function used in the simulations is based on the assumption that sessions running at high rates are able to make more accurate estimations than those running at lower rates. An abstraction of the function itself is made by a number of n bandwidth samples $C_{i,j}$ (cf. Figure 3)

A specific session is then given access to a number of these samples according to its current rate level, and then it will use this as basis for its estimation. A high rate gives a high number of samples available, and then, also, a higher degree of accuracy.

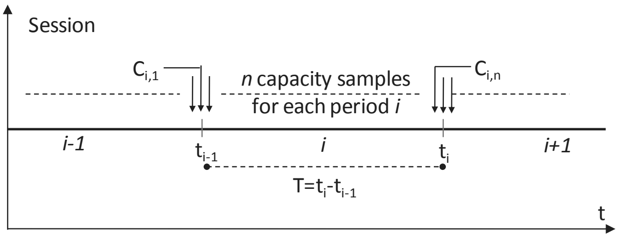


Figure 3. Capacity samples per period

The number of samples $x_{s,i}$ available to a specific session s for period i is given by its ratio between current rate $R_s(t_i)$ and max rate R_smax, multiplied by n as per (2).

$$x_{s,i} = n^{R_s(t_i)}/_{R_smax} \qquad (2)$$

In the simulations, the value of n was set to 20 and R_smax was according to the session definition 5000Kbps. The available bandwidth estimated $A_s(t)$ for period i is then given by the following (3).

$$A_s(t_i) = \sum_{l=1}^{x_i} {C_{i,l}}/{x_{s,i}} \qquad (3)$$

By combination with the expression for $SA_s(t)$ it gives the following expression (4).

$$SA_s(t) = \delta \sum_{l=1}^{x_{i-1}} {C_{i,l}}/{x_{s,i-1}} + (1-\delta) \sum_{l=1}^{x_i} {C_{i,l}}/{x_{s,i}} \qquad (4)$$

The value of δ was set to 0.8 as per [3], thus giving most weight to the available bandwidth estimation from the previous period.

3.3 Simulation Tool

The simulator was built using the process oriented Simula [20] programming language and the Discrete Event Modeling On Simula (DEMOS) context class [21].

This programming language is considered as one of the first object oriented programming languages, and remains a strong tool for performing simulations.

4. RESULTS

The simulation results are presented for different congestion levels on the access link. The chosen capacities are 10, 20, 30 and 40Mbps. The lowest capacity would represent a highly congested scenario. The simulations were also run for all levels from 10 to 40 with increments of 200Kbps but for the sake of clarity these details are left out as they did not change the conclusions.

The studied fairness parameters (proportional and perceived), are compared for the 10 independent users sharing the access link. In order to present more information regarding variations in quality levels, a presentation of Coefficient of Variation (CV) is given. Values for CV below 1 is considered low-variance, while above 1 is considered high-variance.

The simulation results to be presented are based on that all users are accessing the same service, with identical session properties (i.e., quality levels). However, the simulations were also run for other service types and a mixture of services. These results are also left out, as they did not change the conclusion.

4.1 Proportional Fairness

Proportional fairness is measured as achieved session average rate per user, divided by session max – as per the definition earlier (cf. Figure 4, Figure 5, Figure 6). A high value is good and the maximum value is 1.

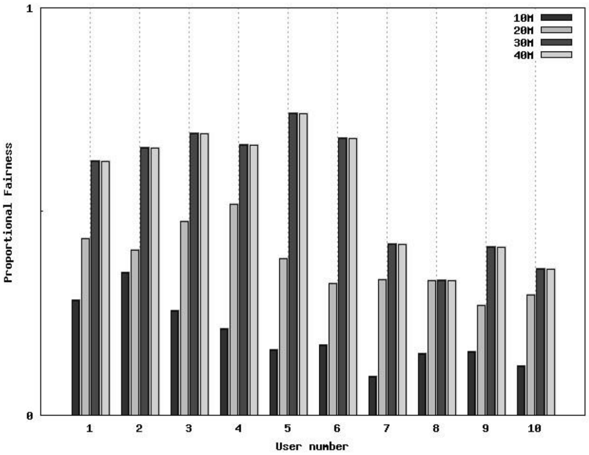


Figure 4. Proportional Fairness, fixed T=2s

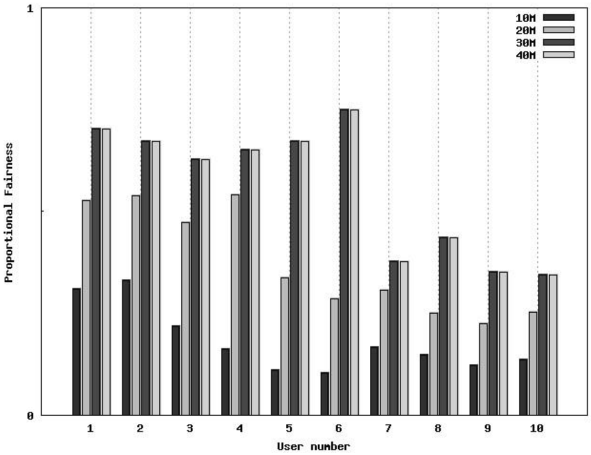


Figure 5. Proportional Fairness, Uniform T [1.6, 2.4]

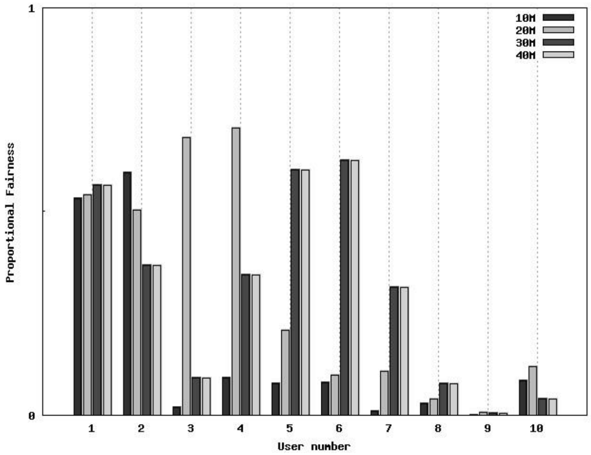


Figure 6. Proportional Fairness, negexp T [λ =0.5]

4.2 Perceived Fairness

The perceived fairness metric is calculated as the number of quality (rate) level reductions per minute (cf. Figure 7, Figure 8, Figure 9). Here, a low metric value is good – as it would reflect less rate reductions per minute.

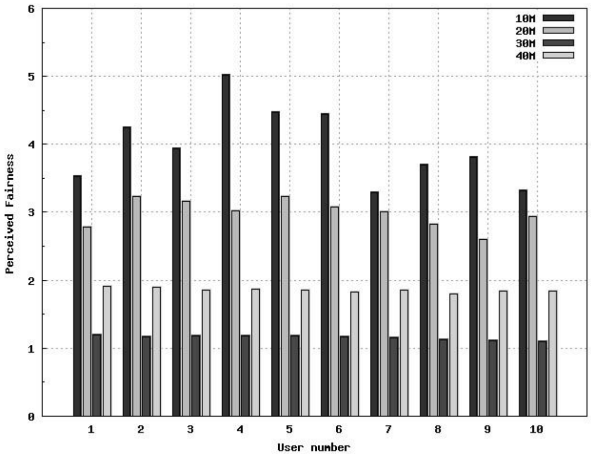


Figure 7. Perceived Fairness, fixed T=2s

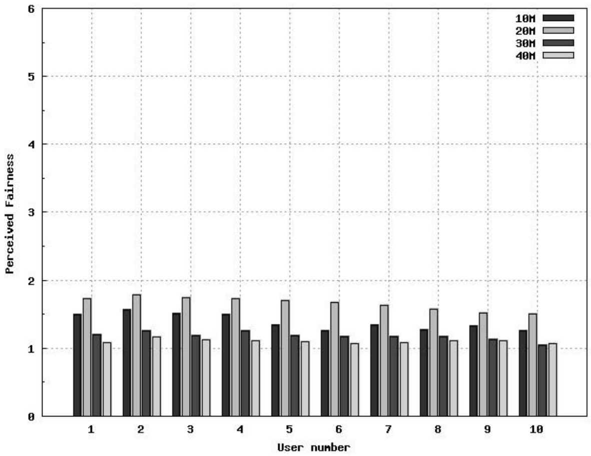


Figure 8. Perceived Fairness, Uniform T [1.6, 2.4]

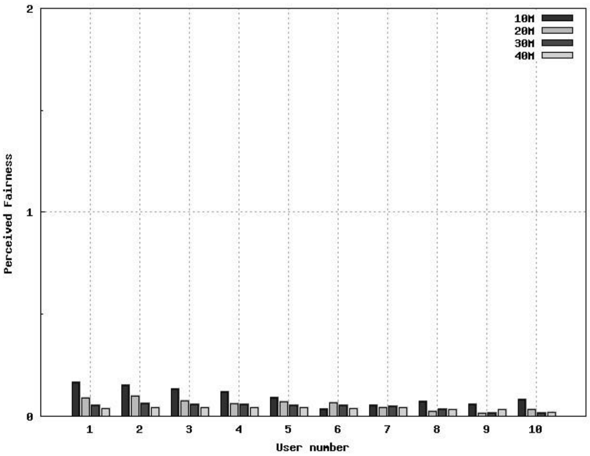


Figure 9. Perceived Fairness, negexp T [λ =0.5]

4.3 Coefficient of Variation (CV)

The Coefficient of Variation is calculated as Standard Deviation/Mean Value for sessions belonging to a user (cf. Figure 10, Figure 11, Figure 12). Values below 1 indicate low-variance which, is preferred.

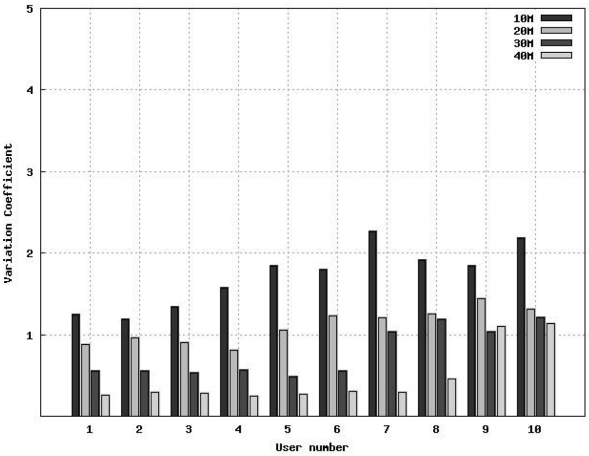


Figure 10. Coefficient of Variation, fixed T=2s

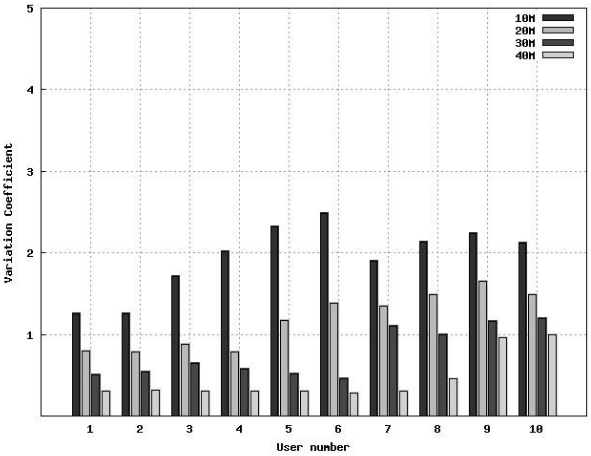


Figure 11. Coefficient of Variation, Uniform T [1.6, 2.4]

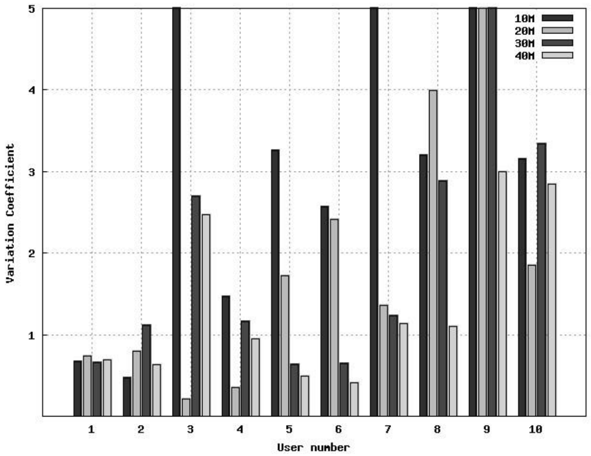


Figure 12. Coefficient of Variation, negexp T [λ =0.5]

5. ANALYSIS

As expected, the randomization of time interval duration does have an effect on the parameters studied. However, the effect is not always positive.

Concerning proportional fairness, the introduction of a uniform *T* variable does not have a significant effect. The result can be viewed as neutral. On the other side, when the negexp *T* variable is used a clear negative effect is observed as the difference between users becomes significant.

For the perceived fairness metric, both the use of a uniform *T* and a negexp *T* have a significant positive effect. The best results are achieved for the negexp case which, gives values well below 1 for all congestion levels and users. It may be considered promising that the effect is especially strong during high times of high congestion (link capacity of 10M and 20M).

Regarding Coefficient of Variation, the results are similar to Proportional Fairness. A uniform *T* give no change, while a negexp *T* gives a negative change.

Summary of simulation results

	Proportional Fairness	***Perceived Fairness***	***Coefficient Variation***
uniform T	neutral	positive	neutral
negexp T	negative	positive	negative

The somewhat intuitive explanation to why changes could be expected is that some of the negative effects of a fixed adjustment interval as illustrated in Figure 13 are reduced. In the case of fixed periods, each session would get the same periodic view on the link utilization, always missing or including some other traffic. This gives a certain degree of error in the available bandwidth estimation functions.

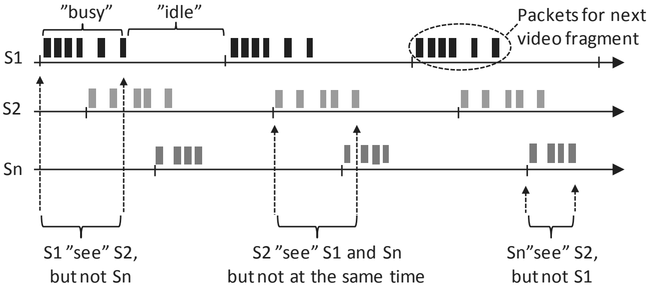

”busy”
”idle”
Packets for next video fragment
S1
S2
Sn
S1 ”see” S2, but not Sn
S2 ”see” S1 and Sn but not at the same time
Sn”see” S2, but not S1

Figure 13. Problem with fixed estimation periods

Each session estimates available bandwidth only during its busy periods (ref Section III Subsection C). This means that in order to get an accurate estimation it is beneficial for it to have overlapping busy periods with as many other sessions as possible.

5.1 Burst Period Duration

The duration of the busy period for a specific session depends on both its current rate level and the rate adjustment interval. The dependency of the rate level follows from the obvious relation to data volume to be transferred per time unit for a specific rate level, while the dependency of rate adjustment interval follows from the requirement to maintain the same average amount of data received over time.

At the beginning of each interval the client requests the next video fragment for a specific rate level, with duration equal to its rate adjustment interval. This is illustrated in Figure 14 where two sessions running at the same rate level, but with different rate adjustment intervals have different busy period durations.

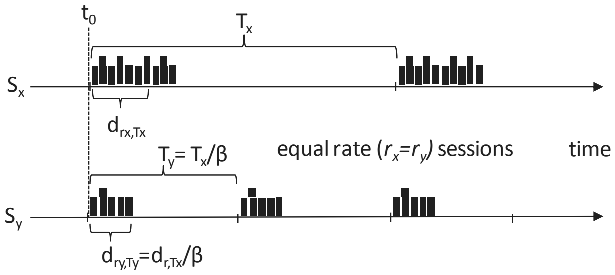


Figure 14. Equal rate sessions with different busy periods

Any two sessions (S_x, S_y) running at the same rate level, will have a relation between their burst period durations expressed by the parameter β. This parameter is given by the following expression (5).

$$\beta = {}^{d_{rx,Tx}}\!/_{d_{ry,Ty}} = {}^{T_x}\!/_{T_y}, \qquad for\, r_x = r_y \qquad (5)$$

Using this relationship, we can express (6) the burst period duration $d_{ri,Ti}$ for any session S_i as a function of its rate adjustment interval T_i and a reference burst period duration $d_{ri,T}$.

$$d_{ri,Ti} = d_{ri,T}({}^{T_i}\!/_{T}) \qquad (6)$$

The values for $d_{ri,T}$ can presumably be calculated based on information about the codec used for the specific media stream inside each sessions, together with assumption on per session server side capacity. Alternatively one could make measurements on a specific system and establish a $d_{ri,T}$ matrix for all valid values of r_i and the reference T value.

However, if we assume that the server side capacity is not a limitation, and that it will always try to burst with a certain bitrate C_{burst} we can also express the burst period duration $d_{ri,Ti}$ as follows (7).

$$d_{ri,Ti} = \left({}^{r_i\,T}\!/_{C_{burst}}\right)\left({}^{T_i}\!/_{T}\right) = \left({}^{r_iT_i}\!/_{C_{burst}}\right) \qquad (7)$$

The maximum value for C_{burst} is natural to think of as the access capacity for the user group / home network, as this is normally the end-to-end bottleneck. However, it is likely that the actual C_{burst} is related to the maximum rate for the specific service.

5.2 Probability for Burst Period Overlap

For T_i values according to a uniform distribution, the probability $P_{i,r,t}$ for a session i at rate level r to be in its busy period at time t will be according to the following expression (8).

$$P_{i,r,t} = {}^{d_{r,Ti}}\!/_{T_i} = \frac{d_{r,T}\left({}^{T_i}\!/_{T}\right)}{T_i} = {}^{d_{r,T}}\!/_{T} \qquad (8)$$

From this, we see that all sessions at a specific rate level has the same probability of being in its busy period at time *t*. We can then express the probability that all *n* sessions are in their busy period at time *t* as follows (9).

$$P_{all\ busy,t} = \left({}^{d_{r_1,T}}\!/_{T}\right)^{c_1} \ \left({}^{d_{r_m,T}}\!/_{T}\right)^{c_m} \qquad (9)$$

The parameter c_m represents the number of sessions at rate level r_m and the sum of all c_m values equals *n*. From this we see that the probability of any session to see all other sessions during its busy period depends on the session rate level mix, and this probability increases when more sessions are running at high rate levels.

Further on, we recognize that the probability for that a session *i* has an overlap with each of the other sessions sometimes during its busy period T_i is the integral of $P_{all\ busy,t}$ over the period [0, T_i] which, is easily expressed as the constant $P_{all\ busy,t}$ multiplied by T_i.

We then let a specific session mix be described by the vector $R_{mix}=\{r_1,...,r_n\}$, whereas r_i represents the rate level for session *i*. Also, for a specific session *i* let A_i be the group of sessions which, has overlapping busy periods with session *i* at a specific time t_0, and B_i be the group of sessions for which, it did not have an overlap. In the situation where all sessions have the same rate adjustment interval duration T_i, the probability of that session *i* has an overlapping busy period with any of the sessions in group B_i at time t_0+T_i is zero. This leads to that while R_{mix} remains unchanged, the view a specific session has of the total traffic will not change. The system state for session *i* in terms of busy period overlap with other sessions is independent of the state at t_0 and also *t* in general.

In the case where T_i is not equal for all sessions, but instead are chosen according to some stochastic distribution – the group of sessions which, overlap the busy period of session *i* at t_0+T_i is not independent of the state at t_0. If we let C_i denote the sub-group of sessions from B_i which, has overlapping busy periods with session *i* at time t_0+T_i, it can be shown that there is a deterministic relationship between A_i, B_i and C_i.

If we then remember the assumed use of a smoothed average function we see the benefit of this potential additional burst period overlaps in subsequent periods.

5.3 Dynamics in Burst Period Overlap

When the starting times for each session and their respective rate adjustment intervals (T_i) are considered stochastic processes, the sessions will combine in time in different ways. In order to define the deterministic relationship between overlapping busy periods during subsequent intervals, we need to analyze scenarios where sessions with different rate levels and different rate adjustment interval are combined.

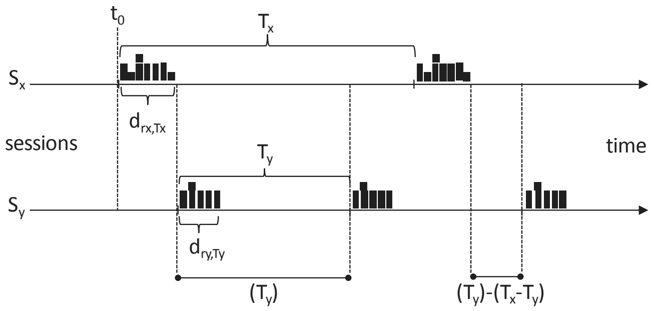


Figure 15. Session Sy staring after Sx (Ty<Tx)

The first scenario (*a*) to be studied is the one where two sessions (S_x, S_y) with different T_i values (T_x, T_y) are active at the same time. We assume $T_x > T_y$ and that S_y starts immediately after the busy period of S_x finishes as illustrated in Figure 15.

For the two sessions (S_x, S_y) there will be shift in phase between them as a function of time which, makes them have a full or partial busy period overlap at some time. The question is then how many rounds it will take for S_x to see S_y and vice versa. It can be shown that we can express the number of rounds for S_x before it has an overlapping busy period with S_y as follows (10).

$$\begin{gathered}
\mathrm{N}_{\mathrm{a,x\rightarrow y}} = 1 + \left\lceil \frac{T_y}{T_x - T_y} \right\rceil \\
\text{when } \frac{T_x}{2} < T_y < (T_x - d_{rx,Tx} - d_{ry,Ty}) \\
\mathrm{N}_{\mathrm{a,x\rightarrow y}} = 2 \\
\text{when } (T_x - d_{rx,Tx} - d_{ry,Ty}) < T_y < T_x
\end{gathered} \qquad (10)$$

In the same way, we can express the number of rounds for S_y before the same overlap of busy period with S_x takes place (11).

$$\begin{gathered}
\mathrm{N}_{\mathrm{a,y\rightarrow x}} = 1 + \left\lceil \frac{T_x}{T_x - T_y} \right\rceil \\
\text{when } \frac{T_x}{2} < T_y < \left(T_x - d_{rx,Tx} - d_{ry,Ty}\right) \\
\mathrm{N}_{\mathrm{a,y\rightarrow x}} = 2 \\
\text{when } \left(T_x - d_{rx,Tx} - d_{ry,Ty}\right) < T_y < T_x
\end{gathered} \qquad (11)$$

The next scenario (*b*) to be studied is where the sessions (S_x, S_y) are running with different T_i values (T_x, T_y) but now S_y finishes its busy period before S_x (cf. Figure 16).

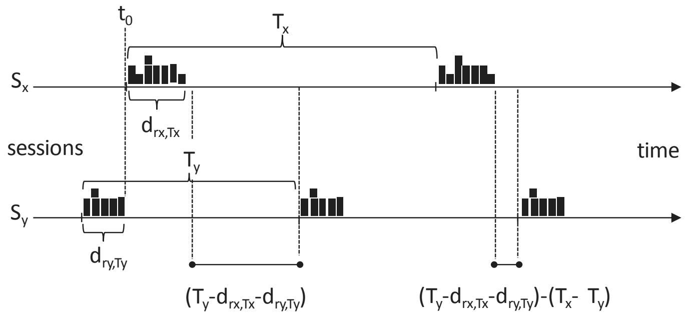


Figure 16. Session Sx staring after Sy (Ty<Tx)

The number of rounds it takes for S_x to see S_y is expressed as follows (12).

$$
\begin{aligned}
& \mathrm{N}_{\mathrm{b,x\to y}} = 1 + \left\lceil \frac{T_y - dx - dy}{T_x - T_y} \right\rceil \\
\text{when} \quad & \tfrac{T_x}{2} < T_y < \left(T_x - d_{r,Ty}\right) \qquad (12) \\
& \mathrm{N}_{\mathrm{b,x\to y}} = 2 \\
\text{when} \quad & (T_x - d_{r,Ty}) < T_y < T_x
\end{aligned}
$$

The number of rounds it takes for S_y to see S_x is expressed as follows (13).

$$
\begin{aligned}
& \mathrm{N}_{\mathrm{b,y\to x}} = 1 + \left\lceil \frac{T_x - dx - dy}{T_x - T_y} \right\rceil \\
\text{when} \quad & \tfrac{T_x}{2} < T_y < (T_x - d_{r,Ty}) \qquad (13) \\
& \mathrm{N}_{\mathrm{b,y\to x}} = 2 \\
\text{when} \quad & (T_x - d_{r,Ty}) < T_y < T_x
\end{aligned}
$$

It should be noted that for both scenarios there is a special case where $N_{a,y\text{-}x}/N_{b,y\text{-}x}$ and $N_{a,x\text{-}y}/N_{b,x\text{-}y}$ are always 2, i.e., two sessions which, did not have overlapping busy periods at t_0 is guaranteed to have overlapped during the next period for S_x and S_y. For a smoothed average function operating over two periods this is desirable, i.e., whatever it does not see in the first period it is guaranteed to see in the next.

5.4 Optimization Problem

The expressions for $N_{y\text{-}x}$ and $N_{x\text{-}y}$ contain many variables. These variables are the rate adjustment intervals T_i and the burst period durations $d_{ri,Ti}$ for all sessions. The latter are calculated based on the session rates r_x and r_y and C_{burst} as defined in Section V. These expressions can be used as input to a constrained optimization problem and analyzed as such in order to find maximum and minimum values.

As the starting point for this optimization problem we can focus on the worst case scenario, that would be the number of rounds for S_y before it has an overlap with S_x ($N_{a,y\text{-}x}/N_{b,y\text{-}x}$), which, will always be higher than the number of rounds for S_x before this has an overlap with S_y.

We also see that $N_{a,y\text{-}x}$ will always be greater than $N_{b,y\text{-}x}$ since $T_x>T_y$. This gives us only one expression to analyze for the worst case scenario as follows (14).

Maximize: $N_{a,y\rightarrow x}$
where

$$N_{a,y\rightarrow x} = \begin{cases} 1+\left\lceil \dfrac{T_x}{T_x - T_y} \right\rceil, if\ T_y < \left(T_x - d_{rx,Tx} - d_{ry,Ty}\right) \\ 2, if\ \left(T_x - d_{rx,Tx} - d_{ry,Ty}\right) < T_y < T_x \end{cases}$$

subject to:

$$1.6 < T_y, T_x < 2.4 \text{ and } T_x/2 < T_y \tag{14}$$

$$r_x, r_y \,\epsilon\, \{250,750,1500,2500,3500,5000\}$$

$$d_{r_x,T_x} = {}^{r_xT_x}\!/_{C_{burst}}$$

$$d_{r_y,T_y} = {}^{r_yT_y}\!/_{C_{burst}}$$

The above maximization can then be done for different values of C_{burst}. In the simulations the access speeds used were between 10 and 40Mbps and the maximum session rate was 5Mbps. Based on measurements of real traffic we can see that the C_{burst} is lower than the actual access speed and thereforee values of respectively 5Mbps, 7.5Mbps and 10Mbps were used for C_{burst}.

For the two different alternatives of choosing values for T_i used in the simulations, the uniform approach is easiest to work with in the optimization context since it gives a min and max value for T_i. For the negexp alternative the corresponding range would be $[0,\infty]$ and for this scenario the optimzation problem does not have a useful solution.

The result from solving the optimization problem is shown in Figure 17. The three different burst bitrates (C_{burst}) give surfaces which, are plotted, whereas the highest capacity gives the highest values for $N_{a,y\text{-}x}$.

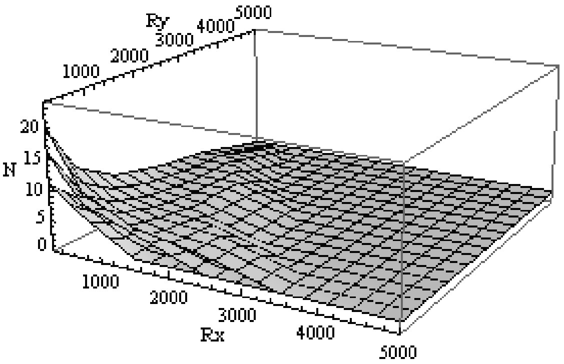


Figure 17. Maximum Na,y-x for different burst bitrates

We see that in many cases we get an overlap already in the second round, and thereby we improve the basis for the available bandwidth estimation algorithm. This analysis then strengthens the findings of the simulations.

The more likely explanation to the negexp behavior in terms of good perceived fairness is the somewhat extreme proportional unfairness. By allowing some sessions to be very greedy, one prevents others from increasing at all. This is a stable but proportionally very unfair situation.

In order to improve the available bandwidth estimations further one may consider the well-known PASTA principle [22] from queuing theory which, states that a Poisson based Arrival process See Time Averages. This implies that the bandwidth probing should take place not only during the burst periods, but as a process taking samples throughout the whole rate adjustment period.

6. CONCLUSIONS AND FUTURE WORK

The results show that there is a significant potential of improving perceived fairness as defined and associated QoE for adaptive streams of the category studied. The positive effect of the suggested enhancement to the rate adaption scheme, i.e., using a stochastically determined duration of rate adjustment intervals rather than fixed values is supported by the simulation results and theoretical analysis.

The results also illustrate that when studying the performance of adaptive streaming solutions, it is not enough to only focus on the network centric QoS domain. A change in this domain does not necessary lead to a corresponding change in the QoE domain, and vice versa. The significant improvement in Perceived Fairness, while proportional fairness remained the same for the uniform T case supports this statement.

As future work in this field it is planned to further study objective and no-reference based QoE metrics such as Perceived Fairness which, is possible to correlate over to the QoS and network domain. It is also planned to verify the simulation and analytical results by means of measurements.

7. ACKNOWLEDGEMENTS

The reported work is done as part of the Road to media-aware user-Dependant self-aDaptive NETWORKS - R2D2 project. This project is funded by The Research Council of Norway. The work has also been actively supported by TV2, the leading commercial TV broadcaster in Norway. TV2 is among the pioneers in providing a full commercial TV offering over the Internet based on ABR technology.

References

[1] A. Zambelli, “IIS smooth streaming technical overview,” http://www.microsoft.com/silverlight/whitepapers/, Tech. Rep., March 2009 (last accessed 10.01.2012).

[2] E. Areizaga, L. Perez, C. Verikoukis, N. Zorba, E. Jacob, and P. Odling, “A road to media-aware user-dependent self-adaptive networks,” in Proc. IEEE Int. Symp. BMSB ’09, 2009, pp. 1–6.

[3] S. Akhshabi, A. C. Begen, and C. Dovrolis, “An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http,” in ACM Multimedia Systems (MMSys), 2011.

[4] L. De Cicco and S. Mascolo, “An experimental investigation of the akamai adaptive video streaming,” in Proceedings of the 6th international conference on HCI in work and learning, life and leisure: workgroup human-computer interaction and usability engineering, ser. USAB’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 447–464.

[5] R. Kuschnig, I. Kofler, and H. Hellwagner, “An evaluation of tcp-based rate-control algorithms for adaptive internet streaming of h.264/svc,” in MMSys ’10. New York, NY, USA: ACM, 2010, pp. 157–168.

[6] B. J. Villa and P. E. Heegaard, “Monitoring and control of QoE in media streams using the click software router,” in NIK2010, Norway. ISBN 978-82-519-2702-4., vol. 1, November 2010, pp. 24–33.

[7] B. J. Villa and P. E. Heegaard,, “Towards knowledge-driven QoE optimization in home gateways,” ICNS 2011, May 2011.

[8] S. Bhatti, M. Bateman, and D. Miras, “Revisiting inter-flow fairness,” in Broadband Communications, Networks and Systems, 2008. BROADNETS 2008. 5th International Conference on, sept. 2008, pp. 585 –592.

[9] S. Floyd, “Metrics for the Evaluation of Congestion Control Mechanisms,” RFC 5166 (Informational), Internet Engineering Task Force, Mar. 2008.

[10] R. Jain, D. Chiu, and W. Hawe, “A quantitative measure of fairness and discrimination for resource allocation in shared systems,” DEC, Tech. Rep., 1984.

[11] D. Mitra and J. B. Seery, “Dynamic adaptive windows for high speed data networks with multiple paths and propagation delays,” Computer Networks and ISDN Systems, vol. 25, no. 6, pp. 663 – 679, 1993, high Speed Networks.

[12] Y. Zhang, S.-R. Kang, and D. Loguinov, “Delayed stability and performance of distributed congestion control,” in Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, ser. SIGCOMM ’04. New York, NY, USA: ACM, 2004, pp. 307–318.

[13] Z. Cao and E. Zegura, “Utility max-min: an application-oriented bandwidth allocation scheme,” in INFOCOM ’99, vol. 2, mar 1999, pp. 793 –801 vol.2.

[14] F. Kelly, “Charging and rate control for elastic traffic,” European Transactions on Telecommunications, 1997.

[15] S. Kunniyur and R. Srikant, “End-to-end congestion control schemes: utility functions, random losses and ecn marks,” IEEE/ACM Trans. Netw., vol. 11, pp. 689–702, October 2003.

[16] A. Jdidi and T. Chahed, “Flow-level performance of proportional fairness with hierarchical modulation in ofdma-based networks,” Comput. Netw., vol. 55, pp. 1784–1793, June 2011.

[17] H. Levy, B. Avi-Itzhak, and D. Raz, “Network performance engineering,” D. D. Kouvatsos, Ed. Berlin, Heidelberg: Springer-Verlag, 2011, ch. Principles of fairness quantification in queueing systems, pp. 284–300.

[18] W. Sandmann, “Quantitative fairness for assessing perceived service quality in queues,” Journal Operational Research, pp. 1–34, April 2011.

[19] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, “Bandwidth estimation: metrics, measurement techniques, and tools,” Network, IEEE, vol. 17, no. 6, pp. 27 – 35, nov.-dec. 2003.

[20] R. J. Pooley, An Introduction to Programming in Simula. Blackwell Scientific Publications. ISBN: 0632014229, 1987.

[21] G. Birtwistle, Demos - A system for Discrete Event Modelling on Simula, G. Birtwistle, Ed. School of Computer Science, University of Sheffeld,, July 1997.

[22] R. W. Wolff, “Poisson Arrivals See Time Averages,” Operations Research, vol. 30, no. 2, pp. 223–231, 1982.

PAPER 5: Improving Fairness for Adaptive HTTP Video Streaming

Bjørn J. Villa, Poul E. Heegaard, Anders Instefjord

Published in

Proceedings of 18th EUNICE/ IFIP WG 6.2, 6.6 International Conference 2012 (EUNICE 2012)

Springer, ISBN 978-3-642-32807-7. pp. 183-193

Aug. 29-31, 2012, Budapest, Hungary

Improving Fairness for Adaptive HTTP Video Streaming

Bjørn J. Villa, Poul E. Heegaard, Anders Instefjord

Abstract: This paper presents an analysis of a suggested method for improving fairness among competing adaptive HTTP video streams. The metrics used as fairness indicators are differences in achieved average rate and stability among the competing streams. The method analyzed is based on changing a fixed and equal video segment request rate of each stream, to either a per session unique or random request rate. The analysis is done by means of measurements in a controlled environment using the Microsoft Smooth Streaming solution. The findings are considered very positive as they show that it is possible to achieve a significant improvement in fairness by applying the suggested method. The main novelty of the paper is that it demonstrates the potential of achieving such improvements without modifying either client or server algorithms.

Keywords: Fairness, Adaptive Streaming, Available Bandwidth Estimation

1. Introduction

Internet services with video components have become a great success in terms of general usage and also as facilitator of new business models on the Internet. A significant part of these services is carried as best-effort traffic between a content provider and the end user. In order to make this possible in a way which is acceptable to the user, a new concept of adaptive video streaming has emerged. The concept is used in solutions from many vendors (Microsoft, Apple, Adobe et.al.) – and they all share the same characteristic in terms of being able to adapt the video quality level during the session according to certain observed metrics (e.g. available bandwidth and CPU load). The main rationale behind this is to enable a video session to survive the varying conditions experienced on the Internet.

Even though the adaptive streaming solutions in the marketplace today share some important characteristics, they remain highly proprietary awaiting the successful completion and adoption of the ongoing active standardization effort known as MPEG DASH [1]. The scope of the standard covers interfaces and message formats, which will enable future interoperability between client and servers from different vendors. Although the new standard will resolve the important interoperability issue, it will not remove the competitive differences between the solutions offered which reflects their approach and set of priorities. As an example, one could look at the solutions from Apple and Adobe which to some extent represent two opposites [2]. The Apple solution tries to maintain a more stable quality (minimize number of changes), at the expense of average quality level. The Adobe solution does it quite differently and almost continuously adjusts the quality level, giving a higher average quality level but at the expense of a potentially higher number of quality changes. The Microsoft solution [3] is somewhere between these two and tries to achieve stability and also a high average quality level, and this is the one used as reference point for the work reported in this paper.

1.1 Problem statement

As the concept of adaptive streaming is becoming widely adopted for delivery of video services on the Internet there are weaknesses appearing. In fact, one could say that the success of the concept becomes its biggest problem. Reason being that the concept works best in a scenario when you are primarily concerned with one video service, and the rest of the traffic is of less importance. However, when multiple adaptive streaming sessions appear at the same time – competing for the same access capacity their performance becomes quite questionable. Traditional topics like intra flow fairness become an issue as well as session stability.

The most likely reason for performance degradation under such conditions is that the algorithms controlling the streams were not made with this in mind. An important area where improvements could be made is with regard to how each session estimates available bandwidth. Ideally, each session should in some way be able to get a notion of all other traffic and thereby also an accurate view on available bandwidth. A candidate reason for why this is not working properly in current solution may be the strict periodic burst/idle nature of each adaptive streaming session (cf. Figure 1).

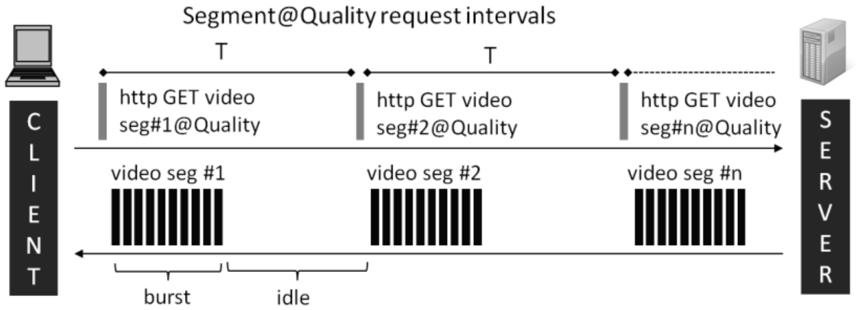

Segment@Quality request intervals
T
T
C L I E N T
http GET video seg#1@Quality
http GET video seg#2@Quality
http GET video seg#n@Quality
S E R V E R
video seg #1
video seg #2
video seg #n
burst
idle

Fig. 1. Burst and idle periods of adaptive http streaming

The periodic characteristic is given by the per session segment@quality request intervals while the burst part is given by the fact that the server always tries to deliver the next segment as fast as possible in order to maintain a high degree of client buffer filling. For the Microsoft Smooth Streaming solution, the typical segment@quality request interval T has a value of 2 seconds. Depending on which quality level a client session is running at it will then be receiving IP packets belonging to the next video segment only during parts of this period T. We call this part the burst period, and the remaining part of T the idle period.

A potential reason for errors made when calculating available bandwidth during each burst period is the fixed and identical periodic behavior of each session. In a situation where the competing traffic is of similar type, e.g. multiple adaptive video streams – we will have that each stream during its burst periods always will notice some of the other streams but at the same time always miss some others. This makes it challenging for the available bandwidth algorithm to provide quality input to the decision making process in the client. The result will be that in many cases the client will make a decision to go up in quality level, even though there is not enough available capacity. This again leads to an increased congestion with a potential overall session impact.

1.2 Research approach

A suggested method of improving fairness among competing adaptive video streams was introduced in our previous work [4] and analyzed by means of simulations. This method was based on changing the fixed T parameter to a random T parameter according to some stochastic distribution.

In addition to this, we now also include new version of the method being the case where each competing stream is set to have a fixed but unique T value. The uniqueness applies among the group of streams competing for a specific amount of bandwidth, e.g. simultaneous streams delivered to members of a household. The purpose of both these suggested changes for the T value is to improve the accuracy of the available bandwidth estimation done by each session.

The approach taken in the work reported in this paper is to take the method validation one step further, by means of performing an experimental validation through a series of measurements in a controlled environment.

1.3 Paper outline

The structure of this paper is as follows. Section 2 provides an overview of related work; Section 3 presents the measurement setup; Section 4 provides the results and an analysis; Section 5 provides the conclusions and an outline of future work is given in Section 6. Acknowledgements are given in Section 7.

2. Related work

The amount of research in this field has been growing over the last years in parallel with commercial deployments and an ongoing standardization effort [6]. The most related research is found in the area of streaming performance evaluation and how to enhance key functionality related to session quality control.

In [7] the authors describe and evaluate selected commercial and open source players for adaptive streaming. They also perform an experimental evaluation of their performance. One of their interesting findings is that competing adaptive streams can cause unpredictable performance for each stream, both in terms of oscillations and ability to achieve fairness in terms of bandwidth sharing. In [8] the authors have investigated how well adaptive streaming performs when being subject to variable available bandwidth in general. Their findings were that the adaptive streams are performing quite well except for during some transient periods. These findings do not contradict the findings in [7] as the type of background traffic used did not have the adaptive behavior itself. In [2] the authors have performed an experimental evaluation of commercial adaptive streaming players in a challenging emulated 3G network environment. Their findings were that there are significant differences between the players in terms of behavior, which supports the findings in [7] but under different network conditions.

In [9] a rate adaptation algorithm which uses a smoothed HTTP throughput measure based on the segment fetch time is proposed and analyzed by means of simulation. This work was taken further by the same authors in [10] and then focusing on the importance of the video segment length and potential of choosing more optimal values for this. Further on, the same research group investigated the potential in using a parallel HTTP streaming approach [11] in order to provide a higher quality. Their simulation results demonstrated a significant potential in doing so. The authors of [12] have also suggested an enhanced method for performing available bandwidth estimation and verified it through experiments. The principle applied is the same as in [9] in the sense that measurements are done based on the data received for each video segment in the stream.

While many publications focus on how to make the client side of an adaptive streaming session more intelligent, the authors of [13] has taken to opposite approach and suggested to make the server side of it more intelligent. Their approach is intended to maintain a low complexity in the player, while at the same time achieving a high degree of effectiveness in session control. Alternative approaches are described in [14] and [15] where the role of the home gateway is investigated in terms of ability to improve adaptive streaming performance

3. Measurement setup

In order to perform the required measurements a testbed was established in a controlled environment including all required components. As illustrated in Figure 2 a number of clients behind an access congestion point were set up to access a Microsoft Smooth Streaming service. The clients were accessing the same adaptive stream from the server and run in a loop with intervals of 25 minute active streaming and then 5 minute break. For each scenario studied the loop was set to give 100 interval repetitions. An earlier developed tool for event reporting [5] from each client (Monitor Plane event reports) is used in order to record interesting events on a per session basis and allow for post processing. The available quality levels for the video stream used were 350, 500, 1000, 1500, 2000, 3000, 4000 and 5000Kbps.

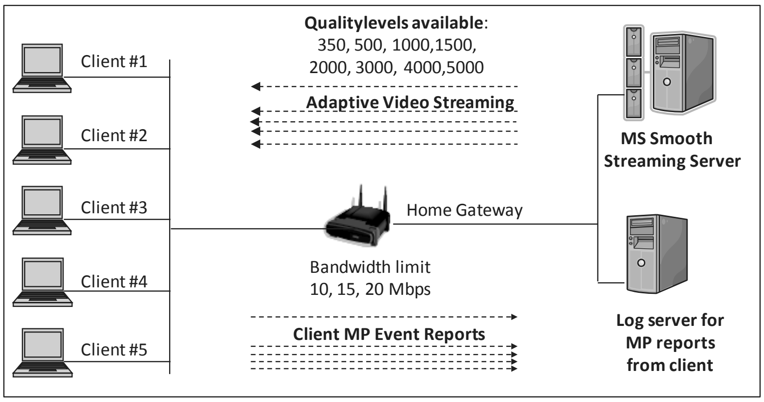


Fig. 2. Measurement testbed

A series of measurement scenarios were defined in order to study the effect of the suggested method under various conditions. The parameters describing a specific scenario was the access capacity shared between the competing adaptive streaming sessions and a specific segment@quality request interval approach (i.e. the T parameter). For the access capacity, values of 10, 15 and 20Mbps were used. This would all be cases with a high degree of congestion as the maximum per session video quality rate was 5Mbps. Thus, all sessions running at max quality rate would not be possible.

With regard to the T parameter, values between 1.6 and 2.5 sec with increments of 0.1 sec were available for use. This differs to some extent from the scenario studied in previous simulations [4] where increments of 0.01sec were available, but was the closest approximation the equipment used allowed for. The following cases were then considered, whereas the first is the reference point (Microsoft Smooth Streaming).

Table 1. Video segment request intervals used in experiments

Fixed T	T=2 sec	default value
Fixed T	T=1.6 sec	Lower than default
Fixed T	T=2.4 sec	Higher than default
Unique T	T_1=1.6, T_2=1.8 T_3=2.0 T_4=2.2 T_5=2.4	Each session different
Random T	T_i=Random{1.6-2.5}, steps of 0.1	Random select

In order to support the objective of validating the effect on fairness among the competing streams, both per session average rate and average number of rate reductions per minute were collected.

4. Results

The results are presented using a graphical view of the sample five-number summaries: the smallest observation, lower quartile, median (and mid 50% samples), upper quartile, and largest observation. The samples come from the measurement intervals as defined in section 3.

The samples for each interval are sorted and then grouped across all intervals according to their order. Statistics are then computed for a specific metric based on the grouped samples. This gives a view on the studied metrics to appear in a sorted manner from the worst to the best performing session, which implicitly also presents the differences between the sessions in a good way.

4.1 Average session rate

The distribution of average rate samples for the 5 competing sessions in the case of changing the fixed T value from its default value, to a higher or lower is presented in Figure 3. The shared access capacity is 10Mbps.

The results show that the effect of changing the fixed T value to a lower or higher value does not have significant effect on the average session rate distributions. The sessions appear to spend most of their time at quality levels 1500Kbps/2000Kbps, and the best/worst performers are approximately within the same bounds as for the default case (T=2.0).

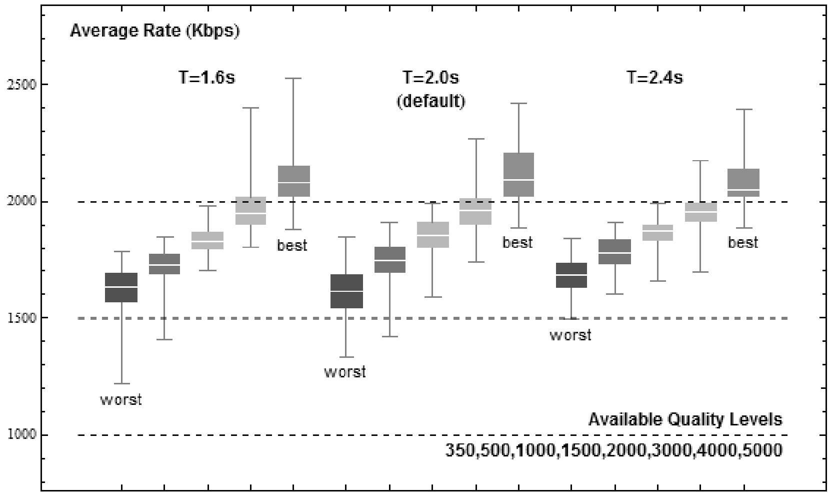


Fig. 3. Average Rate at 10Mbps and T=1.6/2.0/2.4

The distribution of average rate samples for the competing sessions in the case of changing the fixed (and equal) T value from its default value, to either a unique T value or a random T value per session is presented in Figure 4.

The results show that a unique or random T value give distributions for the best performers with lower medians and for the worst performers it gives higher medians. The spread of the median part of the distributions are also smaller. Considering also the outliers in the distributions, it appears that the unique T is better than the random T approach.

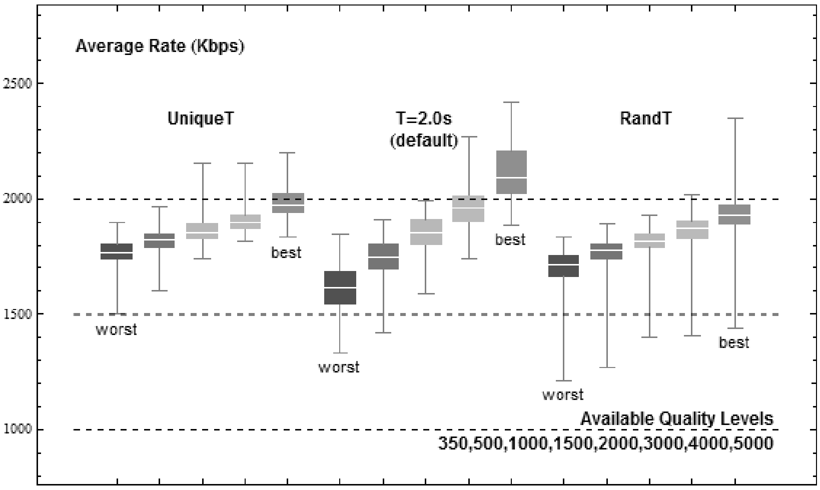


Fig. 4. Average Rate at 10Mbps and T= Unique/2.0/Random

4.2 Session quality reductions

The distribution of quality reduction samples for the competing sessions in the case of changing the fixed T value from its default value, to a higher or lower value is presented in Figure 5.

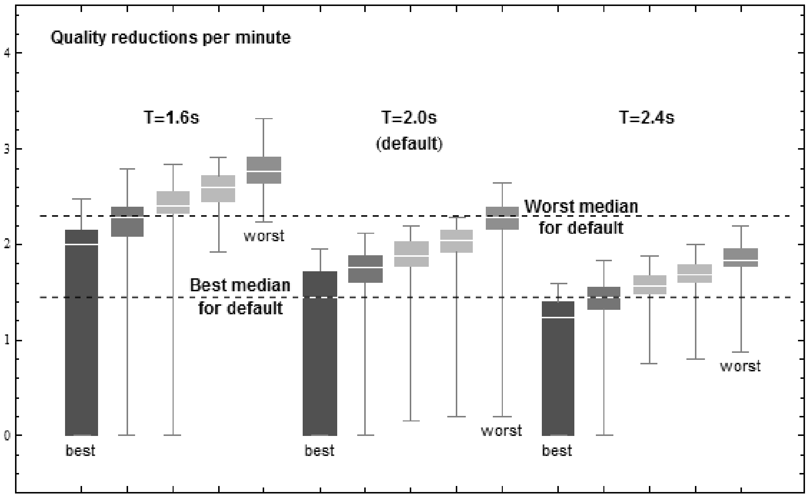


Fig. 5. Quality Reductions at 10Mbps and T=1.6/ 2.0/ 2.4

The results in Figure 5 show that changing the fixed T value to a lower or higher value has a logical effect on the frequency of quality reductions. With a lower T value, we get more reductions and with a higher T value we get less. This makes sense as the lower T value sessions have the opportunity to change quality level more frequently than higher T value sessions.

The distribution of quality reduction samples for the competing sessions in the case of changing the fixed (and equal) T value from its default value, to either a unique T value or a random T value per session is presented in Figure 6.

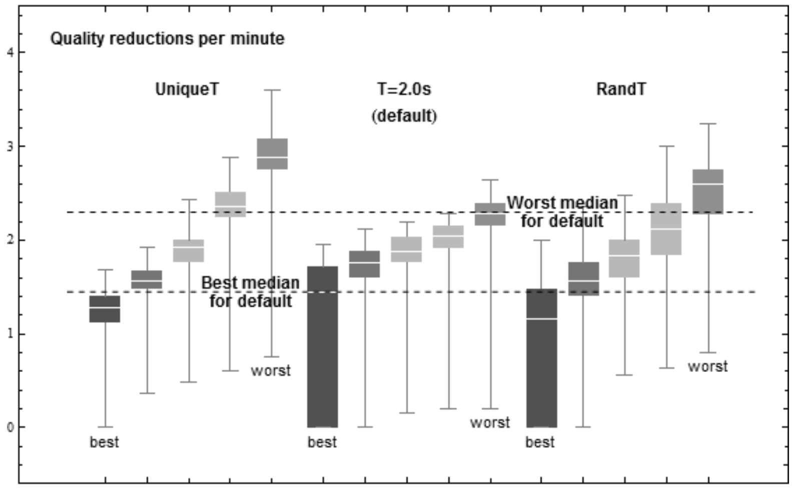


Fig. 6. Quality Reductions at 10Mbps and T= Unique/2.0/Random

The results in Figure 6 show that the method of either using a unique or random T value give distributions for the best performing sessions with lower medians, and higher medians for the worst performing ones. The difference in medians for the best and worst performing sessions is higher for both the unique T and random T cases. However, the absolute values are small. In all cases, the number of quality reductions per minute is mainly between 1 and 3. Thus, the change is not considered significantly negative.

4.3 Session fairness

The two metrics studied in the measurements are both relevant measures for intra-session fairness. The average rate metric is the traditional approach, where the optimal case is considered as the one where each session gets an equal share of the available bandwidth. The quality reduction metric is not so commonly used as a measure of fairness, but we have presented this as a candidate component of perceived fairness in our earlier work [4]. Based on this, we use both these metrics as basis for our evaluation of whether our suggested method of improving fairness for competing adaptive HTTP video streams is effective.

The measurement presented in section 4.1 and 4.2 were for the specific access capacity of 10Mbps as the shared resource for which the sessions as competing. In the following we present combined results including the access capacity levels of 15Mbps and 20Mbps.

In Figure 7 an average rate fairness view is given by the spread of medians for each method (Unique T, Fixed T, Random T) at the three different access capacity levels. It is desirable to have a small difference between the median of the best performer and the worst performer, and also that they are close to the available quality level representing the optimal fair share. The capacity levels used in the measurements were chosen so that there would always be a valid quality level for the fair share as indicated in the figure.

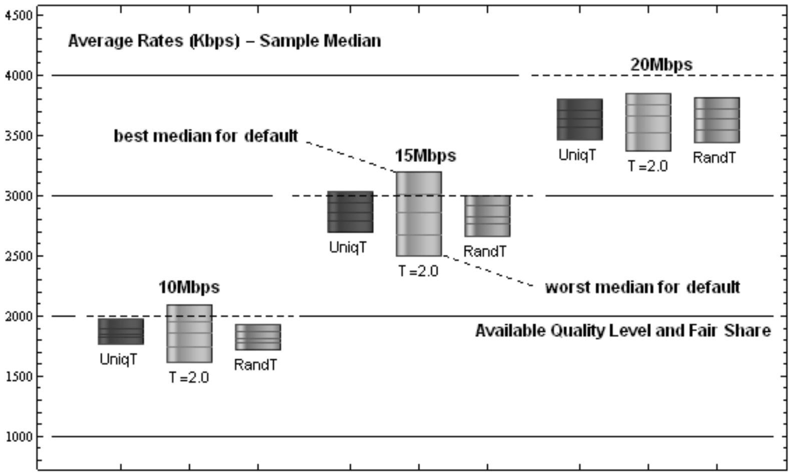


Fig. 7. Average Rate Fairness

The results in Figure 7 show that for both the unique T and random T methods a significant improvement in intra-flow fairness is achieved. The improvement is highest at the two lowest capacity levels where the best/worst performer median difference is reduced by more than 50% while remaining close to the optimal fair share. For the highest capacity level the improvement is lower but still between 20-30%.

Table 2. Quality Reductions view on Fairness

Median	10Mbps Max-Min	15Mbps Max-Min	20Mbps Max-Min
UniqueT	1,60	1,28	1,20
FixedT	0,84	1,08	0,64
RandomT	1,44	1,08	1,04

Looking at fairness among the sessions from a quality reduction per minute point of view, the Max-Min median difference for the unique T, fixed T and random T methods are very close to each other. Therefore, it does not seem as if the significant improvement in average rate fairness come at the expence of stability in general or increased differences among the sessions.

5. Conclusions

The results from the measurements clearly show that the suggested method has a positive effect on fairness among competing adaptive video streams in terms of reducing the difference in average rate per session among competing sessions, without increasing the amount quality reductions significantly. For the two alternative approaches for setting the T value (unique or random) the results are very close to each other, but the unique T approach are slightly better and at the same time a less demanding approach than the random T approach.

A potentially relevant aspect not included as a variable parameter in the measurements is the significance of granularity in video quality levels. Intuitively, having a fine granularity would be ideal – but again this would increase production cost in terms of time and storage requirements.

The value of the findings should be considered most relevant for adaptive HTTP video streaming solutions which embed similar characteristics to the one used in the measurements from Microsoft, but also other adaptive services with a periodic and burst oriented nature. In terms of applying the findings to a real life solution, this is possible to do in different ways. One could of course make all video streams available with a range of different segment sizes and then communicate through the initial session manifest file [3] which one a specific client should use. However, this would be rather demanding from a content storage perspective. Alternatively, one could apply different segment sizes for different video streams and then get much of the same effect assuming that competing streams are not the exact same video stream. This is considered as a likely scenario for users in a home network, where users typically are somewhat different in terms of preferred content. This approach also has a benefit in terms of that it does not require duplication of the content and thereby no increase in storage requirements.

6. Future work

For further enhancements of the performance for competing adaptive HTTP video streams it is likely that one should consider making available bandwidth estimations not only based on the data sent during each burst period, but also have some probing done during the idle periods. Looking closer into the potential effect of introducing some degree of memory in the algorithms controlling the adaptive streams may also provide interesting results. These issues are part of our planned future work, as well as making an integrated analysis of the suggested method using updated results from the earlier developed simulation model [4] together with measurement results

7. Ackowledgements

The reported work is done as part of the Road to media-aware user-Dependant self-aDaptive NETWORKS - R2D2 project. This project is funded by The Research Council of Norway.

References

[1] ISO: Dynamic adaptive streaming over http (dash). iso/iec fcd 23001-6 (2011)

[2] Riiser, H., Bergsaker, H.S., Vigmostad, P., Halvorsen, P., Griwodz, C.: A comparison of quality scheduling in commercial adaptive http streaming solutions on 3G network. In: Proceedings of the 4th Workshop on Mobile Video. MoVid '12, New York, NY, USA, ACM (2012)

[3] Zambelli, A.: IIS smooth streaming technical overview. Technical report (March 2009, last accessed 10.01.2012)

[4] Villa, B.J., Heegaard, P.E.: Improving perceived fairness and QoE for adaptive video streams. ICNS 2012 (March 2012)

[5] Villa, B.J., Heegaard, P.E.: A monitor plane component for adaptive video streaming. In: NIK2011, Norway. ISSN 1892-0713. Volume 1. (Nov 2011)

[6] Sodagar, I.: The mpeg-dash standard for multimedia streaming over the internet. Multimedia, IEEE 18(4) (april 2011)

[7] Akhshabi, S., Begen, A.C., Dovrolis, C.: An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. In: ACM Multimedia Systems (MMSys). (2011)

[8] De Cicco, L., Mascolo, S.: An experimental investigation of the Akamai adaptive video streaming. In: Proceedings of the 6th international conference on HCI in work and learning, life and leisure: workgroup human-computer interaction and usability engineering. USAB'10, Berlin, Heidelberg, Springer-Verlag (2010)

[9] Liu, C., Bouazizi, I., Gabbouj, M.: Rate adaptation for adaptive http streaming. In: the second annual ACM conference Proceedings of the second annual ACM conference on Multimedia systems - MMSys 11, San Jose, CA, USA New York, New York, USA, ACM Press, ACM Press (2011)

[10] Liu, C., Bouazizi, I., Gabbouj, M.: Segment duration for rate adaptation of adaptive http streaming. In: Multimedia and Expo (ICME), 2011 IEEE International Conference on. (July 2011)

[11] Liu, C., Bouazizi, I., Gabbouj, M.: Parallel adaptive http media streaming. In: Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on. (31 2011-aug. 4 2011)

[12] Mok, R.K.P., Luo, X., Chan, E.W.W., Chang, R.K.C.: Qdash: a QoE-aware dash system. In: MMSys. (2012)

[13] De Cicco, L., Mascolo, S., Palmisano, V.: Feedback control for adaptive live video streaming. In: Proceedings of the second annual ACM conference on Multimedia systems. MMSys '11, New York, NY, USA, ACM (2011) 145–156

[14] Houdaille, R., Gouache, S.: Shaping http adaptive streams for a better user experience. In: MMSys. (2012)

[15] Villa, B.J., Heegaard, P.E.: Towards knowledge-driven QoE optimization in home gateways. ICNS 2011 (May 2011)

PAPER 6: Improving Fairness in QoS and QoE domains for Adaptive Video Streaming

Bjørn J. Villa, Poul E. Heegaard

Published in

The International Journal On Advances in Networks and Services,
Volume 5 n. 3&4, 2012

http://www.iariajournals.org/networks_and_services/

IARIA, ISSN: 1942-2644. pp. 291-303

Improving Fairness in QoS and QoE domains for Adaptive Video Streaming

Bjørn J. Villa, Poul E. Heegaard

Abstract: This paper presents an enhancement to a category of Adaptive Video Streaming solutions aimed at improving both Quality of Service (QoS) and Quality of Experience (QoE). The specific solution used as baseline for the work is the Smooth Streaming framework from Microsoft. The presented enhancement relates to the rate adaptation scheme used, and suggests applying a stochastic or fixed/unique setting of the rate adjustment intervals rather than the default fixed/equal approach. The main novelty of the paper is the simultaneous study of both network oriented fairness in the QoS domain and perception based fairness from the QoE domain, when introducing the suggested mechanism. The method used for this study is by means of simulations, measurements and numerical optimization. Perception based fairness is suggested as an objective QoE metric which, requires no reference to original content. The results show that the suggested enhancement has potential of improving fairness in the QoS domain, while maintaining perception based fairness in the QoE domain.

Keywords: Adaptive Video Streaming; Fairness; QoE/QoS

1. Introduction

Solutions for Adaptive Video Streaming are part of the more general concept of ABR (Adaptive Bit Rate) streaming which, covers any content type. The implementation of ABR streaming for video varies between different vendors, and among the more successful one today is the Microsoft Smooth Streaming framework [1]. In general, the different implementations use undisclosed and proprietary functions, even across interfaces between client and server. The latter is addressed by the new MPEG DASH standard [2].

The basic behavior of adaptive video streaming solutions is that the client continuously performs a measurement and estimation of available resources in order to decide which, quality level to request. The relevant resource from the network side is the available capacity along the path between the server and client. Based on this, at certain intervals the client decides to either go up or down in quality level or remain at the current level. The levels are predefined and communicated to the client by the server at session startup. The changes in quality levels are normally done in an incremental approach, rather than by larger jumps in rate level. The rationale behind this is the objective to provide a smooth watching experience for the user. However, it may also be related to the CPU monitoring done by the client, as this is a key resource required. It may be the case that even if the network can provide you with a much higher rate level, the CPU on the device being used would not be able to process it. During the initial phase of an adaptive streaming session the potential requests of change in rate level are more frequent than later on when operating in a more steady-state phase. To some extent this is a rather aggressive behavior from a single client which, may have undesirable inter-stream impacts. At the same time, in order to give the user a good first impression and make him want to continue using the service it is desirable to reach a high quality as soon as possible.

Among the strongest drivers for commercial use of ABR based services on the Internet are Over-The-Top content providers. These are providers which, rely on the best effort Internet service as transport towards their customers. Therefore, technologies aiming at making services survive almost any network state are of great interest. In addition to focus on the network based QoS dimensions of services and involved networks, there is also a growing interest in the QoE dimension [3]. The latter should be considered as not only a richer definition of quality, but also more focused towards who decides whether something is good or bad, i.e., the end user. The evolution of successful services on Internet indicates that the focus on QoE for Over-The-Top providers is a good strategy.

1.1 Problem Statement

The concept of Adaptive Video Streaming is very promising. However, as more and more services are adopting this concept the success brings new challenges. The first challenge with effects visible to the end users is how well these services behave when they compete for a shared resource, such as a home broadband access. With a strong dominance of video based service on the Internet this issue is important to address. As each client operates independently of each other, it has no understanding of the traffic it competes with. Different clients consider each other as just background traffic. This leads to unpredictable behavior of each session. The focus of this paper is to study a method for improving QoS/QoE fairness among competing streams in a home network environment.

1.2 Research Approach

The method investigated in this paper to address the problem at hand is to apply specific changes in the algorithm used by each ABR client controlling the adaptive behavior. The specific change suggested is related to the rate adjustment interval used [1]. The effect of changing the duration of the rate adjustment interval from an equal T duration to either a random or per session unique duration is presented and analyzed.

The ABR solution used as reference point for the work is the one from Microsoft (Smooth Streaming). However, the key principles would still apply to other solutions based on similar principles.

1.3 Paper Outline

The structure of this paper is as follows. Section II presents related work; Section III provides an overview of methodology and metrics; Section IV describes the simulation model; Section V presents simulation results; Section VI presents the measurement setup together with results; In Section VII the simulation results and measurements results are summarized and compared; In Section VII an analytical view of the methods studied are given; Section IX provides the conclusions and an outline of future work.

2. Related Work

It has been shown in [4] that competing adaptive streams can cause unpredictable performance for each stream, both in terms of oscillations and ability to achieve fairness in terms of bandwidth sharing. The experimental results presented give clear indication on that competing ABR clients cause degraded and unpredictable performance. Apart from this paper, the topic at hand does not seem to have been addressed by the academic research community to the extent it deserves.

In another paper [5], the authors have investigated how well adaptive streaming performs when being subject to variable available bandwidth in general. Their findings were that the adaptive streams are performing quite well in this type of scenario except for some transient behavior. These findings do not contradict the findings in [4] as the type of background traffic used do not have the adaptive behavior itself, but is rather controlled by the basic TCP mechanisms.

Rate-control algorithms for TCP streaming in general and selected bandwidth estimation algorithms are described in [6]. This work is relevant to any TCP based application delivering a video stream.

In some of our own previous work we have described and analyzed how competing adaptive streams can be controlled using a knowledge based bandwidth broker in the home gateway [7] [8]. We have also developed a testbed for performing experimental verification of methods studied [9] which has been used for collecting the measurement used in this paper.

3. Methodology and metrics

In this section, we introduce the relevant performance metrics together with motivation for the chosen focus. Thereafter, some candidate methods on how to improve the performance metrics are given, and finally, the specific method subject for study is presented.

3.1 Flow Based Performance Metrics

For transport flows it is common [10] to focus on the following metrics in order to assess their performance: inter-flow fairness, stability and convergence time. This in addition to the general QoS metrics: bandwidth, packet loss, delay and jitter. The same metrics can be applied to adaptive video streams as they by definition also are flows with similar concerns. The analysis of these metrics can be done from a strict network oriented perspective (QoS), but to some extent also bridged over to a user perception domain (QoE). When focusing on the inter-flow fairness metric this is traditionally analyzed [11] using, e.g., the Jain’s fairness index [12], the product measure [13] or Epsilon-fairness [14] for flows with equal resource requirements. For flows with different resource requirement, the Max-Min fairness [15], proportional fairness [16] or minimum potential delay fairness [17] approaches are commonly seen. Real life adaptive video streams would typically belong to the last category.

Max-Min fairness: The objective of max-min fairness is to maximize the smallest throughput rate among the flows. When this is met, the next-smallest throughput rate must be as large as possible, and so on. Max-min fairness can also be explained by considering it as a progressive filling algorithm, where all flows start at zero and grow at the same pace until the link is full. With this approach the max-min fairness gives priority to the smallest flows. The least demanding flows always have the best chance of getting access to all the resources it needs.

Proportional fairness: The original definition of proportional fairness comes from economic disciplines [16] for the purpose of charging. The original definition is used in the relevant RFC [11] but it does not come across as very constructive for the purpose of analyzing fairness in single resource (e.g., bandwidth) sharing among flows. In this context more recent definitions and interpretations are more suitable [18]. The principle of this would be that a resource allocation is considered proportional fair if it is made to the flow which, has the highest ratio between potential maximum resource consumption and its average resource consumption so far. A further simplification would be to use the current resource usage (if greater than 0) instead of the average in the ratio calculation. The same ratio numbers for each flow could then be used to give a view on the current system fairness by comparing them. If they are all equal the system could be stated as proportionally fair.

Minimum potential delay fairness: The idea behind minimum potential delay fairness is based on the assumption that the involved flows are generated by applications transferring files of certain sizes. A relevant bandwidth sharing objective would be to minimize the time needed to complete those transfers. However, this does not apply to an adaptive streaming scenario and is therefore not discussed any further.

3.2 Perception Based Performance Metrics

There is a wide range of metrics which, influence how satisfied an end user is with a service such as e.g., video streaming. Many of these are not related to network aspects, and therefore difficult to influence by means in this domain. However, one of the perceived performance metrics which, could be correlated with network aspect is the notion of perceived fairness. It is then of great interest to try and find methods of influencing this in a positive manner.

Looking at fairness from an end user perception, research from the social science and psychology domain [19] states that this is closely related to what is called 'Social Justice'. In this context a queuing system or any other resource allocation mechanism would be considered as a 'Social System'. It has further been found that users react negatively to any system behavior which, gives better service to other users, unless justification is provided. Such system behavior is considered un-fair, i.e., in violation with the social justice of the system as the end users considers it as discrimination.

The end user notion of system discrimination has been suggested by [20] as an important measure of perceived service quality, and more specifically the perceived fairness is stated to be closely related to the discrimination frequency. It should be noted that analyzing this type of end user perceived discriminations has a challenge in terms of handling the false positive and false negative cases.

Applying the concept of discrimination to competing adaptive streams, it would be related to situations where end user expectations are not met during steady state periods and also negative changes in service delivery during more transient periods. In other words, whatever makes the end user think that he is being discriminated due to other users in the system, will lead to reduced perceived service quality.

In order to use this type of perceived end user discrimination as a measure for how well the algorithm which, controls the adaptive streams are performing, a clear definition regarding what end users are considering as discrimination is required. This could, e.g., be periods with session rate below some threshold, any change in session rate to a lower level or the session rate change frequency.

3.3 QoS and QoE Fairness

Based on the overview given in the previous sections for both flow based and perception based performance metrics, the following definitions are presented for the fairness metrics subject for study in this paper.

In the QoS domain, we use proportional fairness as the key metric while in the QoE domain we use perceived fairness, defined as follows.
Proportional Fairness - The difference between the worst and best performing streaming sessions in terms of average rate achieved during the session lifetime divided by session max rate.

Perceived Fairness – The difference between the worst and best performing streaming sessions in terms of average number of rate reductions (i.e. discrimination events) per minute.

Following this, the main focus is put on differences in performance for the worst and best performing sessions. However, the absolute values for both achieved session rate and session quality level reductions are of course also relevant when evaluating the proposed methods.

3.4 Methods for Improving Performance

There are several things that one could try to incorporate into the adaptive algorithms controlling the ABR service in order to make them perform better in a multi-stream scenario.

The selected performance metrics to be studied are proportional fairness and perceived fairness metric as described. Whether it is possible to improve both these fairness metrics at the same time will be an important part of the results. We consider the following approaches as interesting to consider in this domain.

Randomization or unique time intervals: The equal rate adjustment intervals (*T*) used by each adaptive stream while in steady-state may be a contributing factor to inaccurate estimations of available bandwidth and thereby oscillating behavior. An alternative to fixed intervals would be to randomize them by using a per-session stochastic parameter (within certain reasonable bounds) or assigning each session a unique value. By doing so the available bandwidth estimation methods may become more accurate.

Back-off periods: Whenever a service is reducing its rate level due to observed congestion it may try to increase again after the same amount of time (*T*). In addition to the previous described randomization/unique approach to this interval, one could also consider introducing a back-off period. This would imply that after a service has reduced its rate level, it enters a back-off period of a certain duration during which, no increase is allowed.

Threshold based behavior: Rather than using the same intervals of potential rate changes all the time, one could introduce a threshold for when it operates more or less aggressive. This threshold could be the mean available rate level for a specific session, or even a smoothed average value for the actual achieved level. This concept is applied with success in more recent TCP versions for the purpose of optimizing performance.

The method chosen for this study is according to the first approach described, i.e., using a random or unique interval between each potential rate change. This would represent a different approach than the default method used in Smooth Streaming from Microsoft [1].

As baseline for the simulations, the default interval *T=2s* has been used. Then as alternatives, both a stochastic distribution and per session fixed unique distribution has been implemented. For the stochastic approach the Uniform distribution was chosen with parameters [1.6, 2.4]s. For the fixed unique approach, the sessions were spread on the following value set [1.6, 1.8, 2.0, 2.2, 2.4]s.

4. Simulation model

As the adaptive streaming solutions of today are highly proprietary, the details concerning their implementation are not disclosed. Due to this, there will always be some degree of uncertainty concerning their internal functions.

The simulation model is based on our earlier work [21] but has been somewhat simplified in order to allow for comparison with experimental results.

4.1 Assumptions

One of the key functions of an ABR client is the method used for determining whether to go up or down in rate level during times of varying available bandwidth. From studying live traffic it does not seem as if the clients use additional network probing beyond the actual information obtained through download of video segments. Further on, in the likely absence of a per stream traffic shaper at the server side (for scalability and performance reasons), it will give a traffic pattern for each stream which, typically contains a sequence of burst and idle periods. The measured burst period rate is then higher than the actual stream rate level. Also, it is likely that there will be sub-periods within the burst periods where per packet rate is close to the total available bandwidth. As such, the client can probably obtain a rather accurate indication of maximum available bandwidth by just looking at minimum observed inter-arrival time of packets of known size belonging to the same stream.

However, not all streams will have interleaved burst periods so there is a good chance for each stream to overestimate the potential for additional bandwidth. There is a wide range of bandwidth estimation methods and a few of these are described in [22], but again - as the details of the adaptive streaming solutions are not disclosed we will not discuss this part any further. Independent of which, method being used, there will be some degree of uncertainty which, contributes to variable performance. Further on, we assume the following to be true for the ABR sessions to be studied.

- No stream coordination at server side
- No involvement from mechanisms in the network between the client and server
- All clients operate independently and do not communicate
- All clients are well behaved in the sense that they follow the same scheme
- At each defined stream rate level there are no variations due to i.e., picture dynamics
- All clients access the same stream on the server side

4.2 Session Type and Schedule

The ABR sessions used in the simulator are based on profiles observed in commercial services. The quality levels defined are {0, 350, 500, 1000, 1500, 2000, 3000, 4000, 5000} Kbps. All sessions are of the same type. The sessions are initiated by 5 different users and start time scheduling are done according to the stochastic distributed parameter t_a – Uniform [0, 2000]ms. This gives that all sessions start during the first 2 seconds.

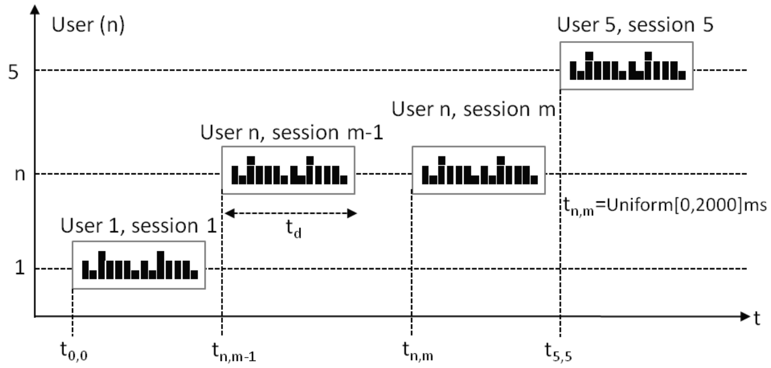


Figure 1. Session scheduling per user

During one simulation run, each user executes a total of 100 sessions sequentially. Time for starting the next session (m) for specific user (n) is noted $t_{n,m}$ (cf. Figure 1). The duration of each session t_d is deterministic and set to 25 minutes.

4.3 Rate Adaptation Algorithm

The model for rate adaptation per session is based on periodic estimation of available bandwidth $A_s(t)$ and calculation of a smoothed average $SA_s(t)$.

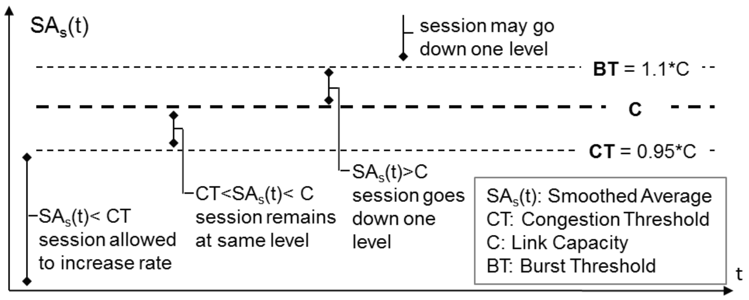

$SA_s(t)$
session may go down one level
BT = 1.1*C
C
CT = 0.95*C
$SA_s(t)$< CT session allowed to increase rate
CT<$SA_s(t)$< C session remains at same level
$SA_s(t)$>C session goes down one level
$SA_s(t)$: Smoothed Average
CT: Congestion Threshold
C: Link Capacity
BT: Burst Threshold
t

Figure 2. Thresholds for smoothed average

This smoothed average (cf. Figure 2) is compared to a congestion threshold (*CT*), the link capacity (*C*) and a burst threshold (*BT*) in order to trigger a rate adjustment.

Whenever the sum of requested rates from sessions is above the burst threshold (*BT*), the next session which, calculates $SA_s(t)$ will be forced down, independent of the value of $SA_s(t)$. This function is implemented in the simulator in order to incorporate the somewhat unpredictable behavior during times of heavy congestion.

The calculation of smoothed average $SA_s(t)$ is based on [4], and is expressed in (1). The parameter δ gives the weighting of the estimated available bandwidth for the two periods included in the calculation.

$$SA_s(t) = \delta\, A_s(t_{i-1}) + (1 - \delta)A_s(t_i) \qquad (1)$$

The available bandwidth estimation function used in the simulations is based on the assumption that sessions running at high rates are able to make more accurate estimations than those running at lower rates. An abstraction of the function itself is made by a number of n bandwidth samples $C_{i,j}$ (cf. Figure 3)

A specific session is then given access to a number of these samples according to its current rate level, and then it will use this as basis for its estimation. A high rate gives a high number of samples available, and then, also, a higher degree of accuracy.

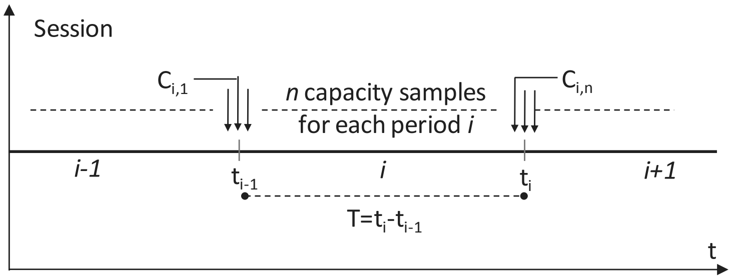


Figure 3. Capacity samples per period

The number of samples $x_{s,i}$ available to a specific session s for period i is given by its ratio between current rate $R_s(t_i)$ and max rate $R_s max$, multiplied by n as per (2).

$$x_{s,i} = n^{R_s(t_i)}/_{R_s max} \qquad (2)$$

In the simulations, the value of n was set to 20 and $R_s max$ was according to the session definition 5000Kbps. The available bandwidth estimated $A_s(t)$ for period i is then given by the following (3).

$$A_s(t_i) = \sum_{l=1}^{x_i} C_{i,l}/_{x_{s,i}} \qquad (3)$$

By combination with the expression for $SA_s(t)$ it gives the following expression (4).

$$SA_s(t) = \delta \sum_{l=1}^{x_{i-1}} C_{i,l}/_{x_{s,i-1}} + (1-\delta) \sum_{l=1}^{x_i} C_{i,l}/_{x_{s,i}} \qquad (4)$$

The value of δ was set to 0.8 as per [4], thus giving most weight to the available bandwidth estimation from the previous period.

4.4 Simulation Tool

The simulator was built using the process oriented Simula [23] programming language and the Discrete Event Modeling On Simula (DEMOS) context class [24].

This programming language is considered as one of the first object oriented programming languages, and remains a strong tool for performing simulations.

5. Simulation results

The simulation results are presented for different capacity levels on the access link. The chosen capacities are 10, 12.5, 15, 17.5 and 20Mbps. At all these capacity levels there would be congestion as the sum of the maximum quality level requested for the 5 competing sessions is 25Mbps. The simulations were also run for capacity levels between those given above, but for the sake of clarity these details are left out as they did not change the conclusions.

Simulation session results are sorted and then grouped according to the studied metrics, giving a clear view on performance ranging from the worst to the best performer.

The characterization is done by looking at the distributional properties location (mean), spread (mid 50% values) and high/low 25% results. For this purpose the box and whisker plots are used as they give a compact view of all these properties.

5.1 Proportional Fairness

As defined, proportional fairness is calculated by the achieved session average rate per user, divided by session max – and then a comparison of these values are done for the competing sessions/users. The results from the simulations give 100 independent samples for this metric.

Improvements in proportional fairness are then recognized as reduced difference between the worst and best performing sessions. The results are presented in Figure 4, Figure 5 and Figure 6 showing both the mean values and the spread of the metric sample distributions.

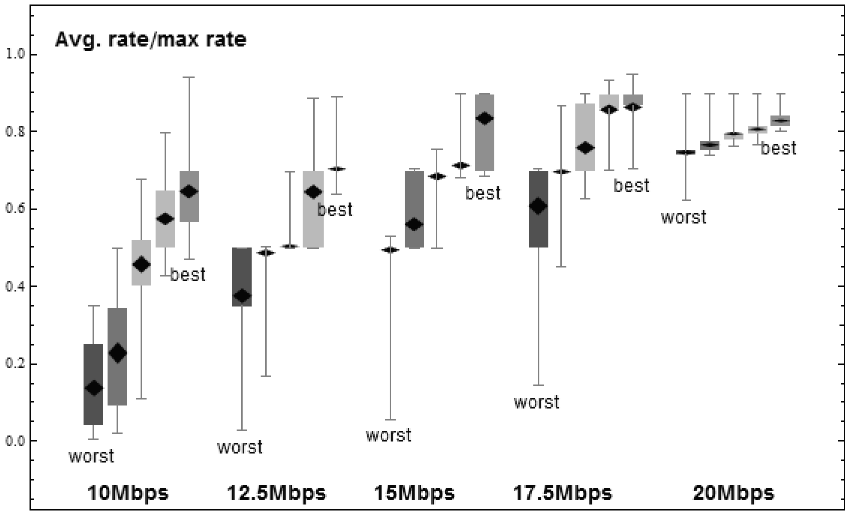


Figure 4. Proportional Fairness, Equal T, Simulations

The results shown in Figure 4 illustrates that there is a significant challenge in terms of proportional fairness when using the default equal T approach for all access capacity levels except for at the highest level (20Mbps).

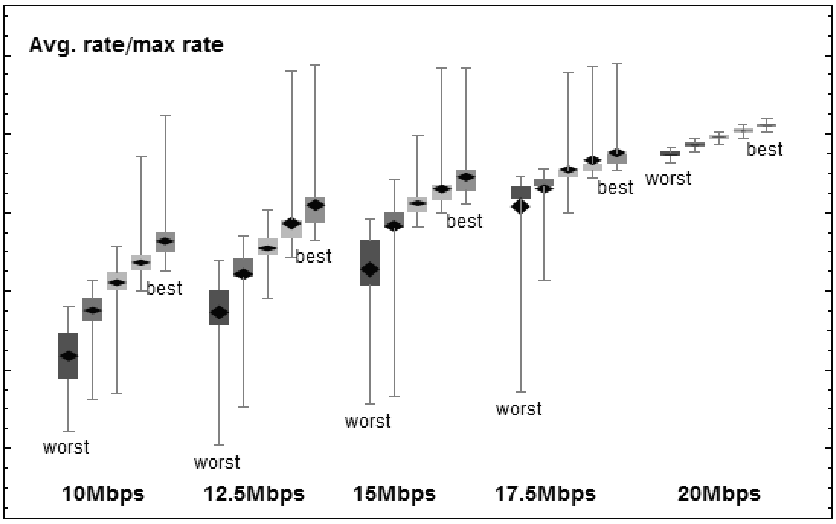


Figure 5. Proportional Fairness, Unique T, Simulations

By careful study of the results shown in Figure 5 for the unique T approach one can see that the difference between the worst and best performing sessions are reduced, and thereby an improved proportional fairness.

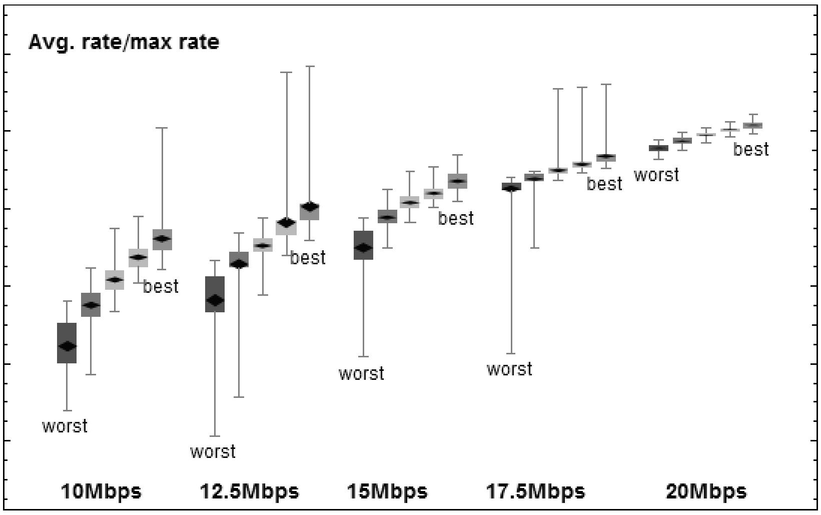


Figure 6. Proportional Fairness, Random T, Simulations

The same effect as for the unique T approach is also visible for the random T approach as shown in Figure 6. For both approaches it is also worth noticing the reduced spread of observations as indicated by the mid 50% values.

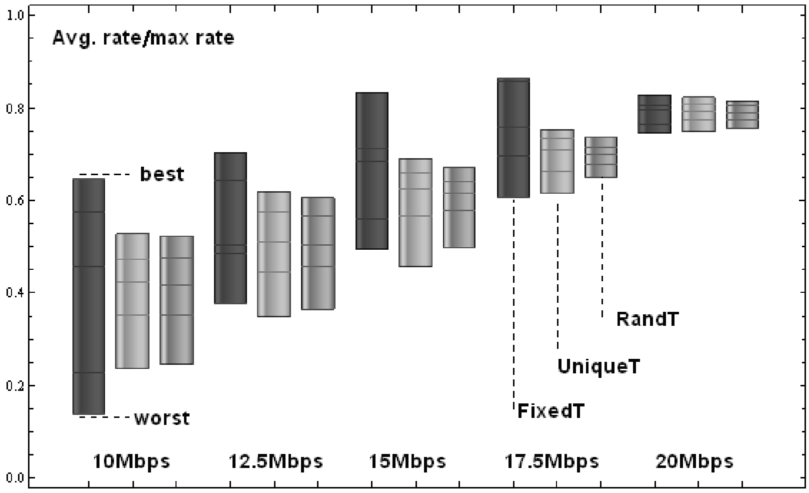


Figure 7. Summary Proportional Fairness, Simulations

In the summary view of the simulations results as shown in Figure 7 the effect of both random T and unique T methods are quite clear. The differences between the competing sessions become smaller, thus we can state that the simulations give reason to believe that the methods studied give improvements in terms of proportional fairness.

5.2 Perceived Fairness

As follows by our definition of perceived fairness a small difference between sessions in terms of number of rate reductions per minute is good. The rationale behind this would be an assumption of that different users have insight into the performance of other sessions. In addition, the absolute value is of course also important. A low metric value is good.

The results shown in Figure 8 give a clear indication on that the simulator model is quite aggressive in terms of how often it allows each stream change its quality level. The level of 15 reductions / minute is likely to represents the model maximum. This follows by T intervals of 2 sec, and our presentation of reductions / minute only.

The results for perceived fairness using the equal T approach are quite poor in the sense that the absolute values are at maximum level for the three mid capacity levels. However, it should be noted that the simulator model contains some simplifications and assumptions which may not be accurate enough in this domain.

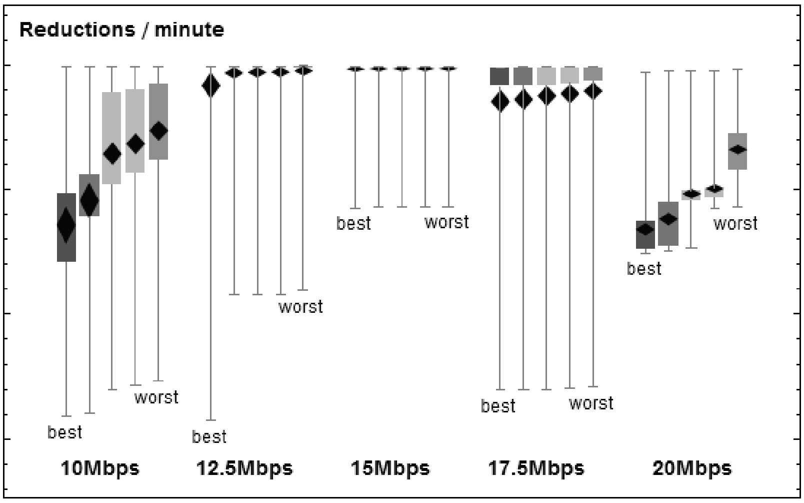


Figure 8. Perceived Fairness, Equal T, Simulations

The results shown Figure 9 for the unique T approach illustrates that the reductions per minute are reduced, but at the same time it introduces a stronger difference between sessions.

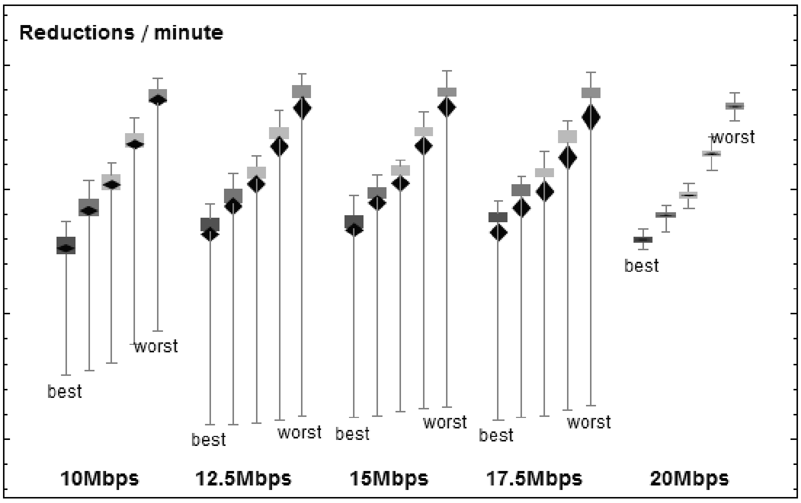


Figure 9. Perceived Fairness, Unique T, Simulations

The same effect as for the unique T approach is also visible for the random T approach as shown Figure 10. Except for the higher spread at 20Mbpss capacity levels, the results are quite similar.

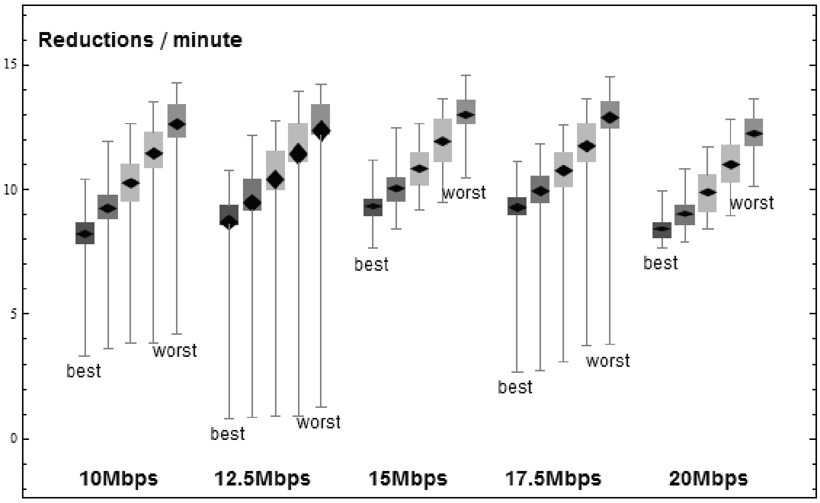


Figure 10. Perceived Fairness, Random T, Simulations

The summary view of perceived fairness is presented in Figure 11. It illustrates both the actual values for the best/worst performers and the difference between them. Results are presented for the default equal T, unique T and random T methods for all access capacity levels. These results alone do not give reason to believe that the investigated method (unique T and random T) give an improved perceived fairness.

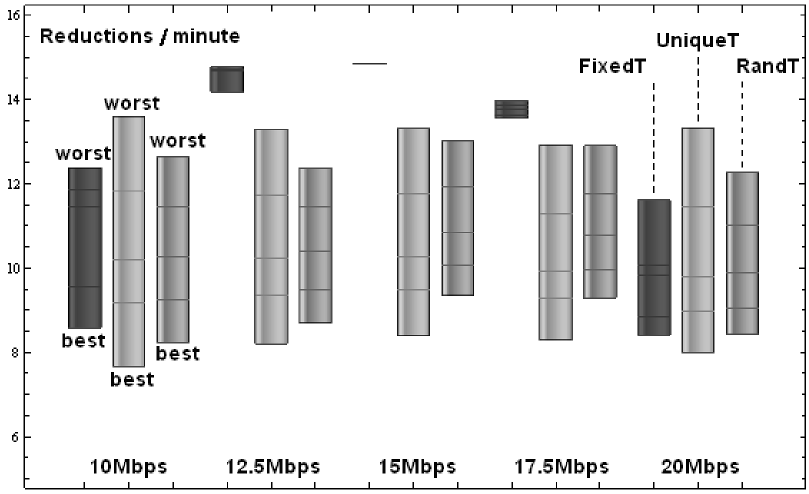


Figure 11. Summary Perceived Fairness, Simulations

6. Measurements

In order to perform the required measurements a testbed was established in a controlled environment including all required component [9]. As illustrated in Figure 12 the 5 clients are located behind a shared access with a certain capacity towards a Microsoft Smooth Streaming service. This scenario matches the one which was built into the simulator as described in section IV.

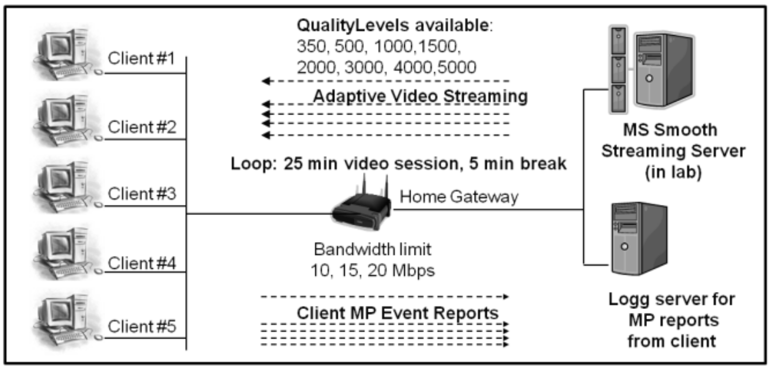


Figure 12. Measurement setup

The clients used were separate PC's with identical HW and SW and set to access the same adaptive HTTP video stream from the lab server. Controlled by scripts on each PC the clients were run in a loop with intervals of 25 minute active streaming and then 5 minute break.

For each scenario studied the loop was set to give 100 interval repetitions. An earlier developed tool for event reporting [25] from each client (Monitor Plane event reports) was used in order to record interesting events on a per session basis and allow for effective post processing.

The measurements results for proportional fairness and perceived fairness are given in the following sections using the same presentation form as for the simulations.

6.1 Proportional Fairness

The results shown in Figure 13 illustrate that there is a problem with regards to proportional fairness when using the default equal T approach for all access capacity levels. The problem is smallest at the highest level (20Mbps), which matches the earlier presented simulation results (cf. Figure 4).

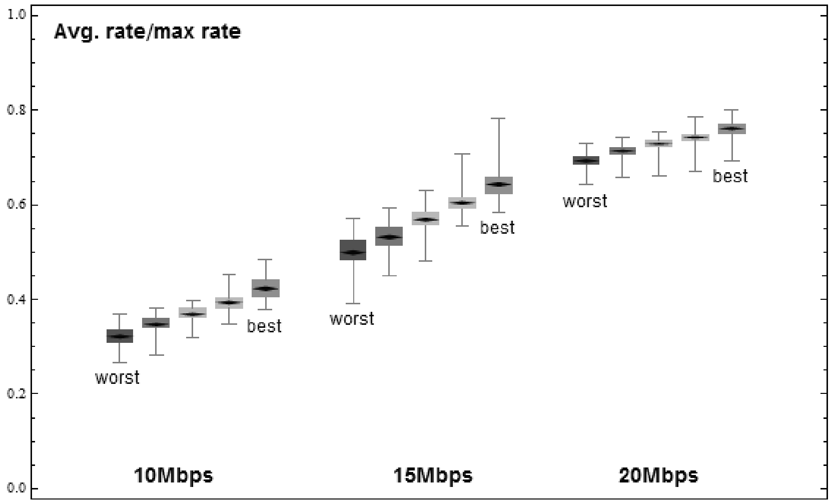


Figure 13. Proportional Fairness, Equal T, Measurements

By studying the results shown in Figure 14 for the unique T approach, a noticeable reduced difference between the worst and best performing sessions are seen. This again, is in line with the corresponding simulation results (cf. Figure 5) indicating improved proportional fairness.

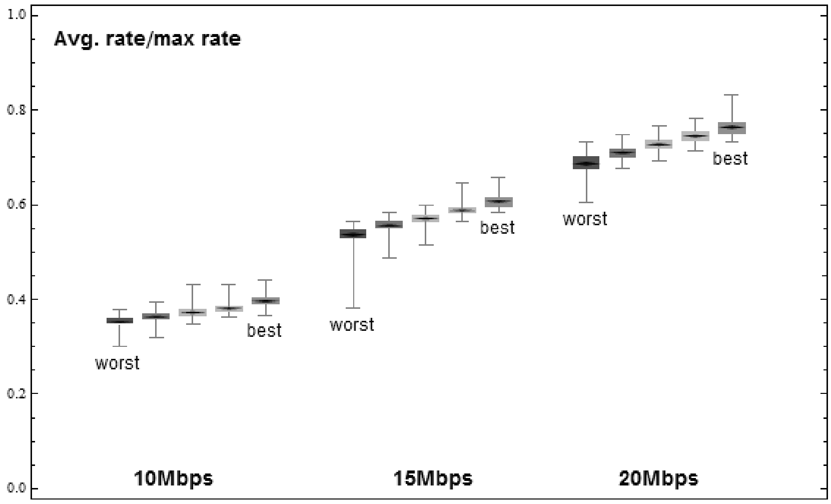


Figure 14. Proportional fairness, Unique T, Measurements

A similar positive effect as for the unique T approach is also visible for the random T approach (cf. Figure 15).

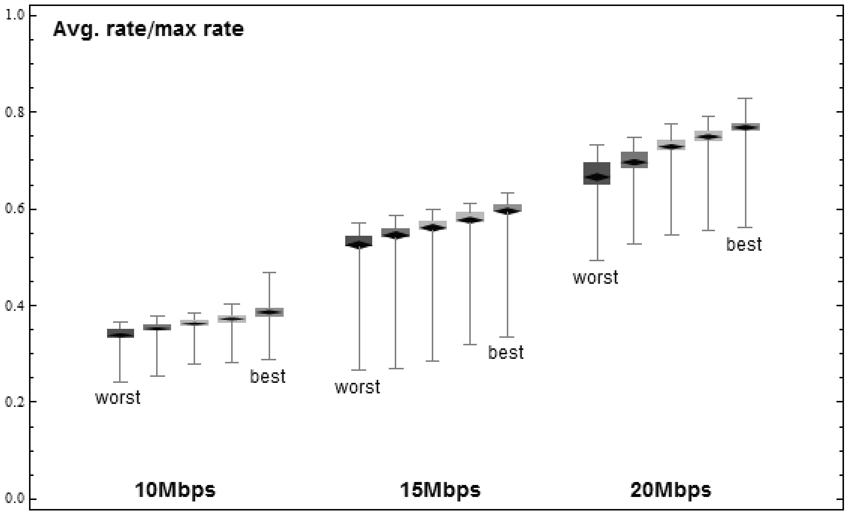


Figure 15. Proportional Fairness, Random T, Measurements

In the summary view of the measurements results as shown in Figure 16 the positive effect of both random T and unique T methods are quite clear, except for at the highest access capacity level (20Mbps). These findings are much in line with the finding from the simulations (cf. Figure 7).

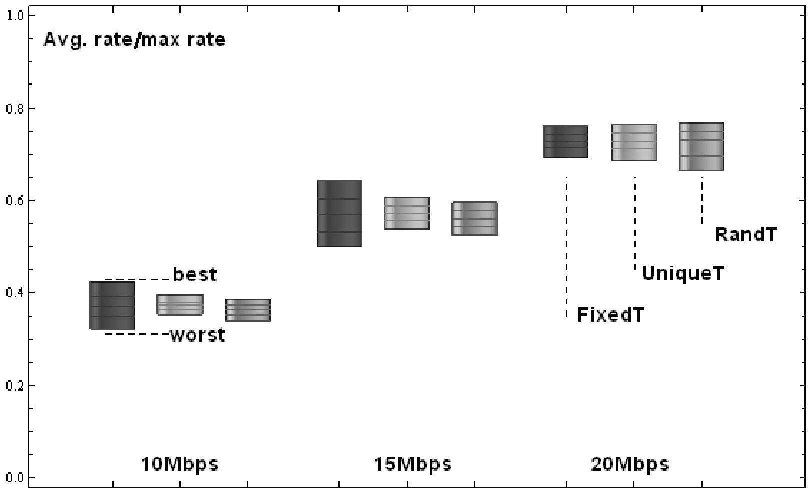


Figure 16. Summary Proportional Fairness, Measurements

It should be noted that measurements were not performed for all the capacity levels which were used in the simulations. The main reason for this is the amount of time required for performing measurements versus time required for simulations.

6.2 Perceived Fairness

The first thing which is noticed when looking at the measurements result for perceived fairness in Figure 17 is that the levels observed are much lower than those collected during simulations (cf. Figure 8). Thereafter, one can see that there is a clear difference between the best and worst performing sessions but the absolute values are rather low.

Therefore, based on these findings we can only state that there is a pure theoretical challenge with perceived fairness. Whether actual users will feel discriminated or get a poor user experience due to quality fluctuations at these levels is not evident.

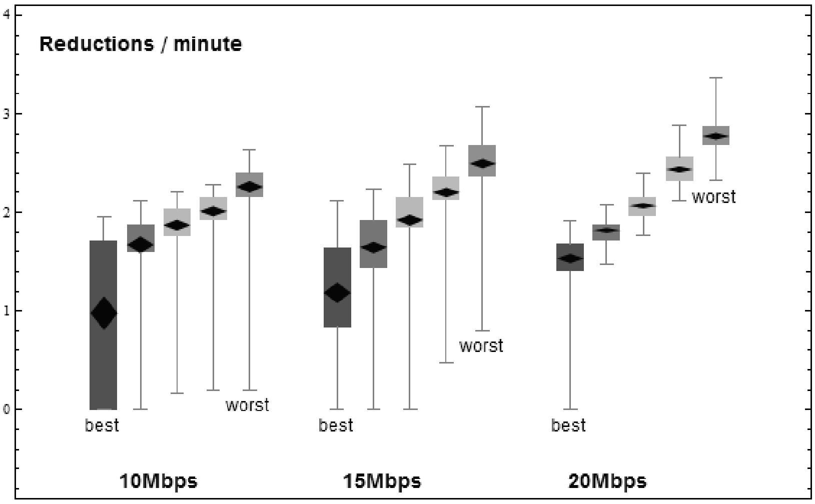


Figure 17. Perceived Fairness, Equal T, Measurements

The results shown Figure 18 for the unique T approach illustrates that the spread in the observations are reduced (mid 50% observations), but the mean value levels remain in the same regions as for the default equal T approach.

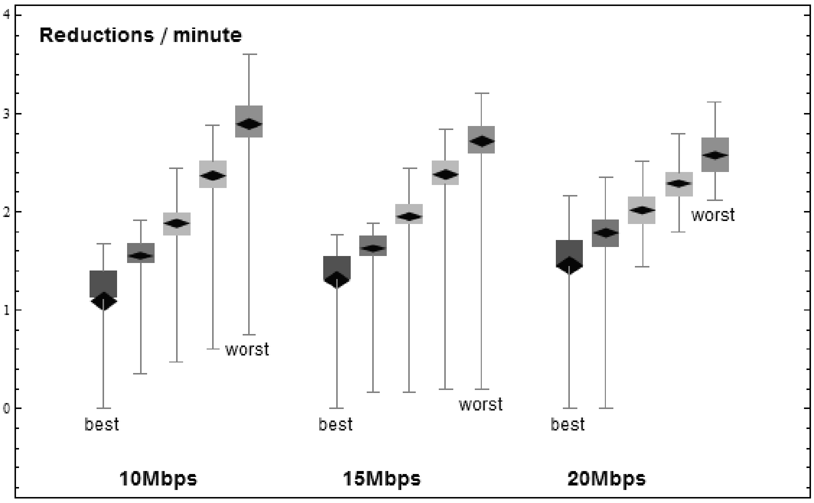


Figure 18. Perceived Fairness, Unique T, Measurements

For the random T approach as illustrated in Figure 19 we see an increase in spread for the observations at the two lowest access capacity levels, making the results in this regard almost similar to the default equal T approach. The exception is the results for 20Mbps access where a quite clear positive effect is seen with regards to difference between the worst and best performing session.

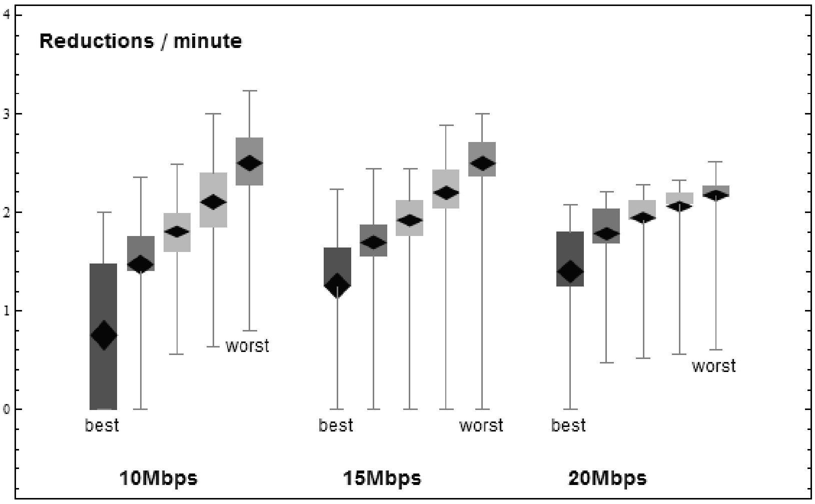


Figure 19. Perceived Fairness, Random T, Measurements

The summary view of perceived fairness based on measurements is presented in Figure 20. As can be seen, the results do not give reason to state an improvement in terms of perceived fairness when implementing either the unique T or random T methods. This is in line with the simulation results, although there is a major difference in the absolute levels.

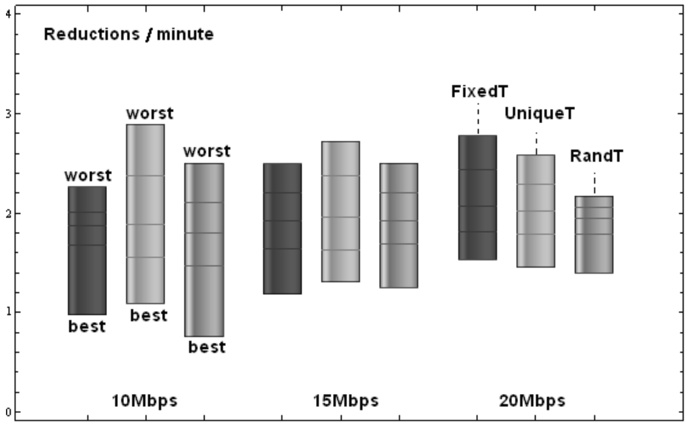


Figure 20. Summary Perceived Fairness, Measurements

7. Comparing simulations and measurements

The results from simulations and measurements differ in absolute values for both proportional fairness and perceived fairness. Keeping in mind that any simulation is based on a model and not the real system itself, this does not come as surprise. However, the important thing is to highlight the effect of introducing the suggested methods (random T, unique T) and see if there are similarities in this regard in both the simulation and measurement domains.

Looking at the combined results for proportional fairness given in Figure 21 we see that a similar effect is present in both domains. There is a clear positive effect of introducing either the random T or unique T method.

Both the simulation results and measurements results show a very strong positive effect for most access capacity levels, except for at the highest level (20Mbps) where the effect is close to neutral.

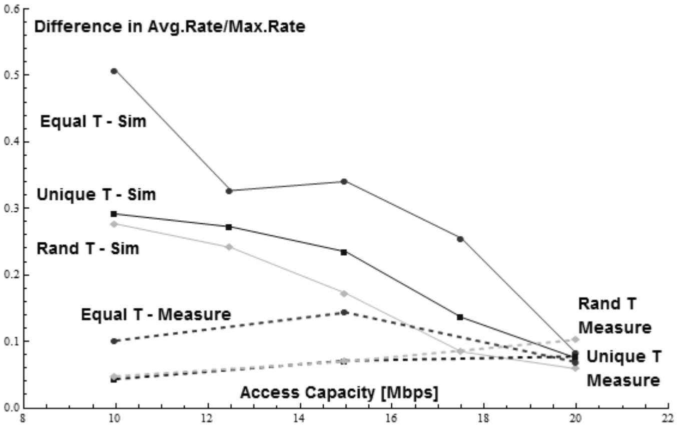


Figure 21. Proportional Fairness, Measurements and Simulations

For the perceived fairness results as shown in Figure 22 the simulation part indicates a strong improvement for the suggested methods. However, as the absolute values are so high (close to assumed maximum) the credibility of these results is weakened. The rate adaptation algorithm implemented in the simulator is probably too aggressive compared to the real life implementations.

The measurement results for perceived fairness are neutral viewed alone, but when combined with the proportional fairness results one can say it is positive that improvements in the pure QoS domain does not come at the expense of degraded performance in the QoE domain.

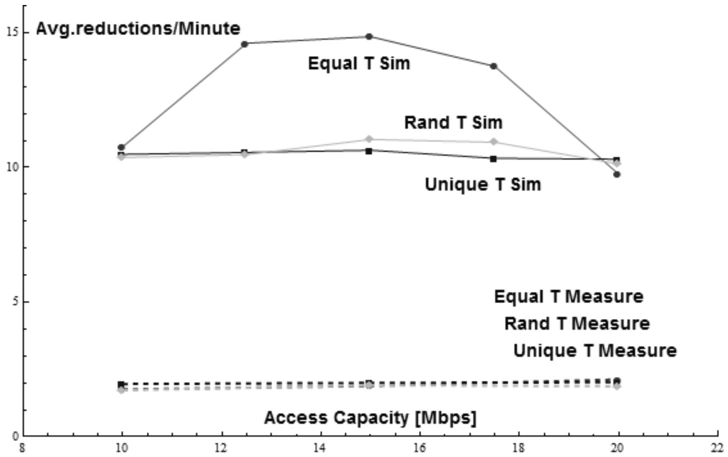


Figure 22. Perceived Fairness, Measurements and Simulations

In summary, the simulations together with the measurements gives a strong indication of that the suggested methods have a potential real life value in terms of improving proportional fairness.

The differences between using a random T value or a per session unique T value does not give basis for saying which is better. However, from an implementation point of view the random approach clearly has its challenges as the video content requires encoding according to these intervals. In light of this, the approach of using per session unique T values is the preferred one.

8. Analysis

The somewhat intuitive explanation to why changes could be expected when introducing either a random T or unique T rate adjustment interval is that some of the negative effects of an equal adjustment interval as illustrated in Figure 23 are reduced. In the case of equal periods, each session would get the same periodic view on the link utilization, always missing or including some other traffic. This gives a certain degree of error in the available bandwidth estimation functions.

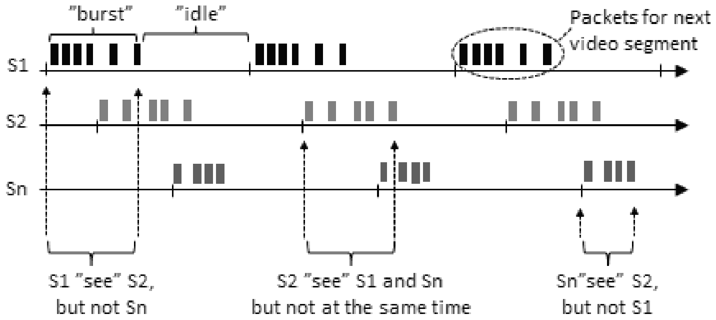

"burst"
"idle"
Packets for next video segment
S1
S2
Sn
S1 "see" S2, but not Sn
S2 "see" S1 and Sn but not at the same time
Sn"see" S2, but not S1

Figure 23. Problem with equal estimation periods

Each session estimates available bandwidth only during its burst periods. Although not explicitly stated in system documentation [1] this has been verified in the measurement testbed [9]. As part of this work the absence of any active network probing was verified. Therefore, whatever notion of available bandwidth each client uses as basis for its rate adaptation it must be based on information collected during the periods where it receives video segments (burst periods). This means that in order to get an accurate estimation it is beneficial for each client to have overlapping burst periods with as many other sessions as possible.

8.1 Burst Period Duration

The duration of the burst period for a specific session depends on both its current rate level and the rate adjustment interval. The dependency of the rate level follows from the obvious relation to data volume to be transferred per time unit for a specific rate level, while the dependency of rate adjustment interval follows from the requirement to maintain the same average amount of data received over time.

At the beginning of each interval the client requests the next video fragment for a specific rate level, with duration equal to its rate adjustment interval. This is illustrated in Figure 24 where two sessions running at the same rate level, but with different rate adjustment intervals have different burst period durations.

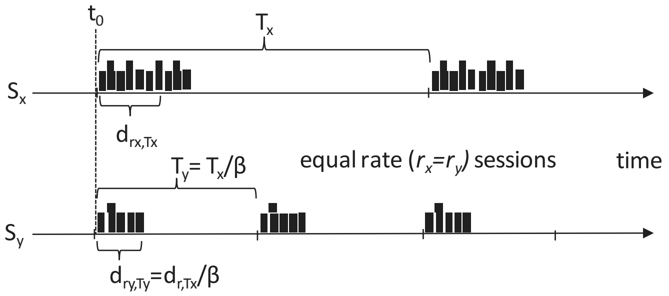


Figure 24. Equal rate sessions with different burst periods

Any two sessions (S_x, S_y) running at the same rate level, will have a relation between their burst period durations expressed by the parameter β. This parameter is given by the following expression (5).

$$\beta = {d_{rx,Tx}}\Big/{d_{ry,Ty}} = {T_x}\Big/{T_y}, \qquad for\ r_x = r_y \qquad (5)$$

Using this relationship, we can express (6) the burst period duration $d_{ri,Ti}$ for any session S_i as a function of its rate adjustment interval T_i and a reference burst period duration $d_{ri,T}$.

$$d_{ri,Ti} = d_{ri,T}({T_i}/{T}) \qquad (6)$$

The values for $d_{ri,T}$ can presumably be calculated based on information about the codec used for the specific media stream inside each sessions, together with assumption on per session server side capacity. Alternatively one could make measurements on a specific system and establish a $d_{ri,T}$ matrix for all valid values of r_i and the reference T value.

However, if we assume that the server side capacity is not a limitation, and that it will always try to burst with a certain bitrate C_{burst} we can also express the burst period duration $d_{ri,Ti}$ as follows (7).

$$d_{ri,Ti} = \left({r_i\,T}\Big/{C_{burst}}\right)\left({T_i}\Big/{T}\right) = \left({r_iT_i}\Big/{C_{burst}}\right) \qquad (7)$$

The maximum value for C_{burst} is natural to think of as the access capacity for the user group / home network, as this is normally the end-to-end bottleneck. However, it is likely that the actual C_{burst} is related to the maximum rate for the specific service.

8.2 Probability for Burst Period Overlap

For T_i values according to a uniform distribution, the probability $P_{i,r,t}$ for a session i at rate level r to be in its burst period at time t will be according to the following expression (8).

$$P_{i,r,t} = {d_{r,Ti}}\Big/{T_i} = \frac{d_{r,T}\left({T_i}\Big/{T}\right)}{T_i} = {d_{r,T}}\Big/{T} \qquad (8)$$

From this, we see that all sessions at a specific rate level has the same probability of being in its burst period at time t. We can then express the probability that all n sessions are in their burst period at time t as follows (9).

$$P_{all\ burst,t} = \left({d_{r_1,T}}\Big/{T}\right)^{c_1} \square \left({d_{r_m,T}}\Big/{T}\right)^{c_m} \qquad (9)$$

The parameter c_m represents the number of sessions at rate level r_m and the sum of all c_m values equals n. From this we see that the probability of any session to see all other sessions during its burst period depends on the session rate level mix, and this probability increases when more sessions are running at high rate levels.

Further on, we recognize that the probability for that a session *i* has an overlap with each of the other sessions sometimes during its burst period T_i is the integral of $P_{all\ burst,t}$ over the period [0, T_i] which, is easily expressed as the constant $P_{all\ burst,t}$ multiplied by T_i.

We then let a specific session mix be described by the vector $R_{mix}=\{r_1,...,r_n\}$, whereas r_i represents the rate level for session *i*. Also, for a specific session *i* let A_i be the group of sessions which, has overlapping burst periods with session *i* at a specific time t_0, and B_i be the group of sessions for which, it did not have an overlap. In the situation where all sessions have the same rate adjustment interval duration T_i, the probability of that session *i* has an overlapping burst period with any of the sessions in group B_i at time t_0+T_i is zero. This leads to that while R_{mix} remains unchanged, the view a specific session has of the total traffic will not change. The system state for session *i* in terms of burst period overlap with other sessions is independent of the state at t_0 and also *t* in general.

In the case where T_i is not equal for all sessions, but instead are chosen according to some stochastic distribution – the group of sessions which, overlap the burst period of session *i* at t_0+T_i is not independent of the state at t_0. If we let C_i denote the sub-group of sessions from B_i which, has overlapping burst periods with session *i* at time t_0+T_i, it can be shown that there is a deterministic relationship between A_i, B_i and C_i.

If we then remember the assumed use of a smoothed average function we see the benefit of this potential additional burst period overlaps in subsequent periods.

8.3 Dynamics in Burst Period Overlap

When the starting times for each session and their respective rate adjustment intervals (T_i) are considered stochastic processes, the sessions will combine in time in different ways. In order to define the deterministic relationship between overlapping burst periods during subsequent intervals, we need to analyze scenarios where sessions with different rate levels and different rate adjustment interval are combined.

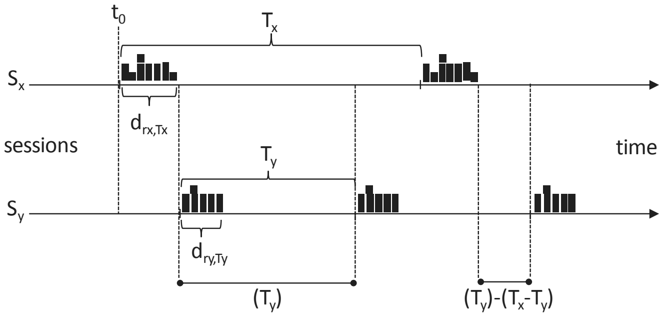


Figure 25. Session Sy staring after Sx (Ty<Tx)

The first scenario (*a*) to be studied is the one where two sessions (S_x, S_y) with different T_i values (T_x, T_y) are active at the same time. We assume $T_x > T_y$ and that S_y starts immediately after the burst period of S_x finishes as illustrated in Figure 25.

For the two sessions (S_x, S_y) there will be shift in phase between them as a function of time which, makes them have a full or partial burst period overlap at some time. The question is then how many rounds it will take for S_x to see S_y and vice versa. It can be shown that we can express the number of rounds for S_x before it has an overlapping burst period with S_y as follows (10).

$$\mathrm{N}_{\mathrm{a,x\to y}} = 1 + \left\lceil \frac{T_y}{T_x - T_y} \right\rceil$$

when $\frac{T_x}{2} < T_y < (T_x - d_{rx,Tx} - d_{ry,Ty})$ (10)

$$\mathrm{N}_{\mathrm{a,x\to y}} = 2$$

when $(T_x - d_{rx,Tx} - d_{ry,Ty}) < T_y < T_x$

In the same way, we can express the number of rounds for S_y before the same overlap of burst period with S_x takes place (11).

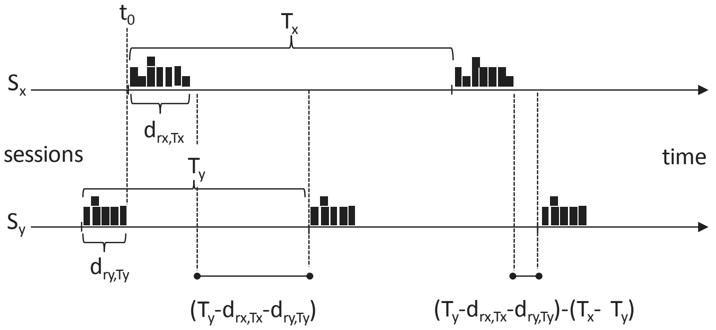
$$\mathrm{N}_{\mathrm{a,y\to x}} = 1 + \left\lceil \frac{T_x}{T_x - T_y} \right\rceil$$

when $\frac{T_x}{2} < T_y < \left(T_x - d_{rx,Tx} - d_{ry,Ty}\right)$ (11)

$$\mathrm{N}_{\mathrm{a,y\to x}} = 2$$

when $\left(T_x - d_{rx,Tx} - d_{ry,Ty}\right) < T_y < T_x$

The next scenario (*b*) to be studied is where the sessions (S_x, S_y) are running with different T_i values (T_x, T_y) but now S_y finishes its burst period before S_x (cf. Figure 26).



t_0
T_x
S_x
$d_{rx,Tx}$
sessions
T_y
time
S_y
$d_{ry,Ty}$
$(T_y$-$d_{rx,Tx}$-$d_{ry,Ty})$
$(T_y$-$d_{rx,Tx}$-$d_{ry,Ty})$-$(T_x$- $T_y)$

Figure 26. Session Sx staring after Sy (Ty<Tx)

The number of rounds it takes for S_x to see S_y is expressed as follows (12).

$$\begin{aligned} &N_{b,x\to y} = 1 + \left\lceil \frac{T_y - dx - dy}{T_x - T_y} \right\rceil \\ \text{when}\quad & \tfrac{T_x}{2} < T_y < \left(T_x - d_{r,Ty}\right) \\ &N_{b,x\to y} = 2 \\ \text{when}\quad & (T_x - d_{r,Ty}) < T_y < T_x \end{aligned} \tag{12}$$

The number of rounds it takes for S_y to see S_x is expressed as follows (13).

$$\begin{aligned} &N_{b,y\to x} = 1 + \left\lceil \frac{T_x - dx - dy}{T_x - T_y} \right\rceil \\ \text{when}\quad & \tfrac{T_x}{2} < T_y < (T_x - d_{r,Ty}) \\ &N_{b,y\to x} = 2 \\ \text{when}\quad & (T_x - d_{r,Ty}) < T_y < T_x \end{aligned} \tag{13}$$

It should be noted that for both scenarios there is a special case where $N_{a,y-x}/N_{b,y-x}$ and $N_{a,x-y}/N_{b,x-y}$ are always 2, i.e., two sessions which, did not have overlapping burst periods at t_0 is guaranteed to have overlapped during the next period for S_x and S_y. For a smoothed average function operating over two periods this is desirable, i.e., whatever it does not see in the first period it is guaranteed to see in the next.

8.4 Optimization Problem

The expressions for N_{y-x} and N_{x-y} contain many variables. These variables are the rate adjustment intervals T_i and the burst period durations $d_{ri,Ti}$ for all sessions. The latter are calculated based on the session rates r_x and r_y and C_{burst} as defined in Section V. These expressions can be used as input to a constrained optimization problem and analyzed as such in order to find maximum and minimum values.

As the starting point for this optimization problem we can focus on the worst case scenario, that would be the number of rounds for S_y before it has an overlap with S_x ($N_{a,y-x}/N_{b,y-x}$), which, will always be higher than the number of rounds for S_x before this has an overlap with S_y.

We also see that $N_{a,y-x}$ will always be greater than $N_{b,y-x}$ since $T_x > T_y$. This gives us only one expression to analyze for the worst case scenario as follows (14).

Maximize: $N_{a,y\rightarrow x}$

where

$$N_{a,y\rightarrow x} = \begin{cases} 1 + \left\lceil \dfrac{T_x}{T_x - T_y} \right\rceil, if\ T_y < \left(T_x - d_{rx,Tx} - d_{ry,Ty}\right) \\ 2, if\ \left(T_x - d_{rx,Tx} - d_{ry,Ty}\right) < T_y < T_x \end{cases}$$

subject to: (14)

$1.6 < T_y, T_x < 2.4$ and $T_x/2 < T_y$

$r_x, r_y \epsilon \{350,500,1000, 1500, 2000, 3000,4000, 5000\}$

$$d_{r_x,T_x} = {r_x T_x}/{C_{burst}}$$

$$d_{r_y,T_y} = {r_y T_y}/{C_{burst}}$$

The above maximization can then be done for different values of C_{burst}. In the simulations and measurements the access capacities used were between 10 and 20Mbps and the maximum session rate was 5Mbps. Based on measurements of real traffic we can see that the C_{burst} is lower than the actual access speed and thereforee values of respectively 5Mbps, 7.5Mbps and 10Mbps were used for C_{burst}.

For the two different alternatives of choosing values for T_i used in the simulations and measurements, both the random T and unique T approaches are possible to work with in the optimization context. However, as the unique T approach will be a special case (subset) of the random T, we only present results for the random T approach.

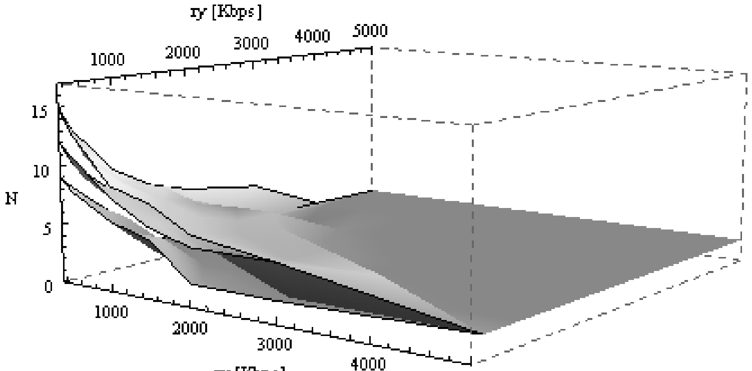


Figure 27. Maximum Na,y-x for different burst bitrates

The result from solving the optimization problem is shown in Figure 27. The three different burst bitrates (C_{burst}) give surfaces which, are plotted, whereas the highest capacity gives the highest values for $N_{a,y\text{-}x}$.

We see that in many cases we get an overlap already in the second round, and thereby we improve the basis for the available bandwidth estimation algorithm. This analysis then strengthens the findings in both the simulations and measurements.

In order to improve the available bandwidth estimations further one may consider the well-known PASTA principle [26] from queuing theory which, states that a Poisson based Arrival process See Time Averages. This implies that the bandwidth probing should take place not only during the burst periods, but as a process taking samples throughout the whole rate adjustment period. However, as this implies some degree of active probing it would potentially have some other undesirable effects.

9. Conclusions and future work

The results show that there is a significant potential of improving proportional fairness as defined while maintaining perceived fairness for adaptive streams of the category studied. The positive effect of the suggested enhancement to the rate adaptation scheme, i.e., using a random or unique duration of rate adjustment intervals rather than the default equal value is supported by simulation results, measurements and also rationalized by the theoretical analysis. The findings differ to some extent from those in our previous work [21], but at the same time we now have a more refined and accurate view of the methods studied. The added value of results from measurements has been significant.

The results also illustrate that when studying the performance of adaptive streaming solutions, it is not enough to only focus on the network centric QoS domain. A change in this domain does not necessary lead to a corresponding change in the QoE domain, and vice versa.

As future work in this field it is planned to further study objective and no-reference based QoE metrics which, is possible to correlate over to the QoS and network domain. It is also planned to study various available bandwidth algorithms with regard to their real-time capabilities and thereby suitability for adaptive video streaming.

10. Acknowledgements

The reported work is done as part of the Road to media-aware user-Dependant self-aDaptive NETWORKS - R2D2 project. This project is funded by The Research Council of Norway.

The work has also been actively supported by TV2, the leading commercial TV broadcaster in Norway. TV2 is among the pioneers in providing a full commercial TV offering over the Internet based on ABR technology.

References

[1] A. Zambelli, “IIS Smooth Streaming Technical Overview,” Microsoft, Tech. Rep., March 2009.

[2] ISO, Dynamic adaptive streaming over HTTP (DASH). ISO/IEC FCD 23001-6, International Organization for Standardization Std. ISO/IEC FCD 23001-6, 2011.

[3] E. Areizaga, L. Perez, C. Verikoukis, N. Zorba, E. Jacob, and P. Odling, “A road to media-aware user-dependent self-adaptive networks,” in Proc. IEEE Int. Symp. BMSB ’09, 2009, pp. 1–6.

[4] S. Akhshabi, A. C. Begen, and C. Dovrolis, “An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http,” in ACM Multimedia Systems (MMSys), 2011.

[5] L. De Cicco and S. Mascolo, “An experimental investigation of the Akamai adaptive video streaming,” in Proceedings of the 6th international conference on HCI in work and learning, life and leisure: workgroup human-computer interaction and usability engineering, ser. USAB’10. Berlin. Springer-Verlag, 2010, pp. 447–464.

[6] R. Kuschnig, I. Kofler, and H. Hellwagner, “An evaluation of TCP-based rate-control algorithms for adaptive internet streaming of h.264/svc,” in MMSys ’10. New York, NY, USA: ACM, 2010, pp. 157–168.

[7] B. J. Villa and P. E. Heegaard, “Monitoring and control of QoE in media streams using the Click Software Router,” in NIK2010, Norway. ISBN 978-82-519-2702-4., vol. 1, November 2010, pp. 24–33.

[8] B. Villa and P. Heegaard, “Towards knowledge-driven QoE optimization in home gateways,” ICNS 2011, Thinkmind 2011, ISBN 978-1-61208-133-5. May 2011, pp. 252-256.

[9] B. J. Villa, P. E. Heegaard, and A. Instefjord, “Improving fairness for adaptive HTTP video streaming,” in EUNICE, 2012, Springer 2012, ISBN 978-3-642-32807-7, pp. 183–193.

[10] S. Bhatti, M. Bateman, and D. Miras, “Revisiting inter-flow fairness,” in Broadband Communications, Networks and Systems, 2008. BROADNETS 2008. 5th International Conference on, sept. 2008, pp. 585 –592.

[11] S. Floyd, “Metrics for the Evaluation of Congestion Control Mechanisms,” RFC 5166 (Informational), Internet Engineering Task Force, Mar. 2008.

[12] R. Jain, D. Chiu, and W. Hawe, “A quantitative measure of fairness and discrimination for resource allocation in shared systems,” DEC, Tech. Rep., 1984.

[13] D. Mitra and J. B. Seery, “Dynamic adaptive windows for high speed data networks with multiple paths and propagation delays,” Computer Networks and ISDN Systems, vol. 25, no. 6, pp. 663 – 679, 1993, high Speed Networks.

[14] Y. Zhang, S.-R. Kang, and D. Loguinov, “Delayed stability and performance of distributed congestion control,” in Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, ser. SIGCOMM ’04. New York, NY, USA: ACM, 2004, pp. 307–318.

[15] Z. Cao and E. Zegura, “Utility Max-Min: An application-oriented bandwidth allocation scheme,” in INFOCOM ’99, vol. 2, mar 1999, pp. 793 –801 vol.2.

[16] F. Kelly, “Charging and rate control for elastic traffic,” European Transactions on Telecommunications, 1997.

[17] S. Kunniyur and R. Srikant, “End-to-end congestion control schemes: utility functions, random losses and ecn marks,” IEEE/ACM Trans. Netw., vol. 11, pp. 689–702, October 2003.

[18] A. Jdidi and T. Chahed, “Flow-level performance of proportional fairness with hierarchical modulation in OFDMA-based networks,” Comput. Netw., vol. 55, pp. 1784–1793, June 2011.

[19] H. Levy, B. Avi-Itzhak, and D. Raz, “Network performance engineering,” D. D. Kouvatsos, Ed. Berlin, Heidelberg: Springer-Verlag, 2011, ch. Principles of fairness quantification in queueing systems, pp. 284–300.

[20] W. Sandmann, “Quantitative fairness for assessing perceived service quality in queues,” Journal Operational Research, pp. 1–34, April 2011.

[21] B. Villa and P. Heegaard, “Improving Perceived Fairness and QoE for Adaptive Video Streams,” ICNS 2012, In Proceedings of ICNS 2012: The Eighth International Conference on Networking and Services, Thinkmind 2012, ISBN 978-1-61208-186-1, March 2012, pp. 149-158.

[22] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, “Bandwidth estimation: metrics, measurement techniques, and tools,” Network, IEEE, vol. 17, no. 6, pp. 27 – 35, nov.-dec. 2003.

[23] R. J. Pooley, An Introduction to Programming in Simula. Blackwell Scientific Publications. ISBN: 0632014229, 1987.

[24] G. Birtwistle, Demos - A system for Discrete Event Modelling on Simula, G. Birtwistle, Ed. School of Computer Science, University of Sheffeld,, July 1997.

[25] B. J. Villa and P. E. Heegaard, “A Monitor Plane Component for Adaptive Video Streaming,” in NIK2011, ISBN 978-8-25192-702-4, November 2011, pp. 145–154.

[26] R. W. Wolff, “Poisson Arrivals See Time Averages,” Operations Research, vol. 30, no. 2, pp. 223–231, 1982.

PAPER 7: Group Based Traffic Shaping for Adaptive HTTP Video Streaming

Bjørn J. Villa, Poul E. Heegaard

Proceedings of the 27th IEEE International Conference on Advanced
Information Networking and Applications (AINA 2013).
IEEE Computer Society Press, ISBN 978-0-7695-4953-8. pp. 830-837
March. 25-28, 2013, Barcelona, Spain

Group Based Traffic Shaping for Adaptive HTTP Video Streaming

Bjørn J. Villa, Poul E. Heegaard

Abstract: Adaptive video streaming is based on the concept of allowing the quality level of a video stream to change during its lifetime based on certain parameters. This approach makes the video stream more tolerant with respect to fluctuations in available bandwidth during the session lifetime. In order for a video stream to decide which quality level to request from the server, it is typical to use both parameters from the client side such as CPU load and estimated parameters from the network side such as available bandwidth. In order for available bandwidth estimations to become more accurate, it is beneficial with some degree of traffic shaping. This paper describes a new method of achieving a traffic shaping effect for adaptive video streaming sessions delivered to members of a specific user group identified by their unique network destination. The novelty of the method is represented by the use of segment size control per video session, rather than active shaping of each session. The effect of the method is analyzed by means of measuring packet IAT and video segment fetch durations, in a controlled lab environment using the Smooth Streaming framework from Microsoft.

Keywords: Adaptive HTTP Video Streaming, Available Bandwidth Estimation, Traffic Shaping

1. Introduction

There are different approaches to adaptive video streaming depending on which technology vendor is being used. The main commercial drivers behind the concept have been Microsoft, Apple, Adobe and Netflix. Their solutions have many similarities, but remain incompatible. The recent MPEG-DASH standard [1] will hopefully resolve this by facilitating a standardized client / server interface.

Common for all adaptive video streaming solutions is that it requires the media content to be available on the server side in different qualities. Each of these quality levels correspond to a certain network level capacity requirement in terms of bits per second. From a quality granularity perspective it would be good to have as many quality levels as possible available on the server, but as this drives the storage requirements it is commonly seen that the number of available quality levels are below 10.

In an area where there are interesting differences between the current commercial solutions is with regard to how often a stream is allowed to adjust its quality level. The solutions from Apple and Adobe represent to some extent two opposites [2]. The Apple solution tries to maintain a more stable quality, at the expense of average quality level.

The Adobe solution does it quite differently and almost continuously adjusts the quality level, giving a higher average quality level at the expense of a potentially higher number of quality changes. The Microsoft solution [3] is somewhere between these two and tries to achieve stability and also a high average quality level. The frequency, of which a quality level change request is allowed, corresponds to the frequency of video segment request. It is therefore convenient to think of these as Segment@Quality requests. The actual size in bytes for each segment depends on both the quality level and the duration of video playback time it shall generate, i.e. the segment duration.

The algorithms controlling whether a streaming session should change its quality level (up or down) or just remain at current level, is outside the scope of the MPEG DASH specification. The details on how this is done in current commercial solutions are not disclosed, but the behavior of the different solutions has been well studied in several papers. These studies indicate that the clients use some kind of measurement or indicator on available bandwidth during the session lifetime as input to their session quality control.

1.1 Problem Statement

Estimating available bandwidth along a network path without having access to statistics from each involved network component represents a challenge in all time dimensions. Methods for doing this has been subject for extensive research for many years, and will be further introduced in section II.

Common for most methods of estimating available bandwidth is that they work best under the fluid flow network traffic assumption. In the fluid flow case, the data packets are evenly spaced according to the average traffic rate. However, in real life – traffic seldom appears according to this assumption.

If we look at traffic originating from adaptive video streaming session, a typical pattern of burst and idle periods are observed (cf. Fig. 1). Based on this we can say that from the perspective of being able to estimate available bandwidth as accurate as possible, it would be ideal if the server side tried to shape the traffic to make it closer to the fluid flow model. However, as this type of functionality is rather demanding to implement on the server side from a pure processing perspective, alternative approaches to achieve similar effect are of interest.

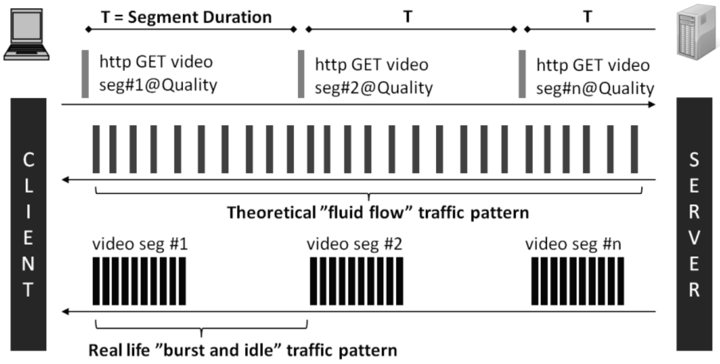

T = Segment Duration
T
T
http GET video seg#1@Quality
http GET video seg#2@Quality
http GET video seg#n@Quality
CLIENT
SERVER
Theoretical "fluid flow" traffic pattern
video seg #1
video seg #2
video seg #n
Real life "burst and idle" traffic pattern

Figure 1. Fluid flow versus burst/idle traffic pattern

The next obvious point where shaping could be applied for a group of users would be at network edge or the home gateway. This approach scales much better, but requires involvement from the network operator, at least for the network edge part. With regards to the home gateway case, it requires some additional functionality and also a permission to use it for this purpose. The latter is not given if the broadband operator owns and operates it.

The problem at hand is then how to achieve a traffic shaping effect for adaptive video streams delivered to a specific user group without actually shaping each of the streams. The method investigated in this paper to achieve this could be viewed as in indirect approach, in the sense that the effect is attempted achieved by controlling how traffic from video streams which are competing for the same bandwidth are interleaved.

1.2 Research Approach

The general motivation for trying to achieve a traffic shaping effect is driven by the likely presence of some kind of available bandwidth estimation algorithm internally in the clients. The rationale for seeking methods which achieve this without active use of network components along the path between client and server is best understood in the context of value chains not involving the network operators. In this context the network operator considers the video services as just any other service carried as part of their best effort traffic class. Despite this, for content providers operating in this domain it is of great interest to find methods to improve the quality of their services. Improving the ability to estimate available bandwidth for adaptive services is considered essential in this regard.

1.3 Paper Outline

The structure of this paper is as follows. Section II provides an overview of related work; Section III presents the traffic shaping method; Section IV presents the measurement setup; Section V provides the results; Section VI provides the conclusions and an outline of future work is given in Section VII.

2. Related work

The amount of research in this field has been growing over the last years in parallel with commercial deployments and an ongoing standardization effort [1]. The performance of existing commercial pre-DASH implementations has been studied, and potential enhancements to the rate adaptation algorithm have been presented. Some examples of this would be the work reported in [4] where a smoothed HTTP throughput measure based on the segment fetch time is suggested as an input metric, and the work in [5] where the focus is on the video segment length and the potential in choosing more optimal values for this. An interesting suggestion was also made in [6] where the measurements of available bandwidth are done based on the data received for each video segment in the stream. Some of this work must of course be understood in the light of that the internal behavior of adaptive video streaming clients remain undisclosed and will not be subject for standardization.

Even though a lot of effort is being put into the specific context of estimating available bandwidth for adaptive video services, it does not mean that the topic itself is a new one. The discipline of estimating available bandwidth has been subject for a lot of research the last years, and a lot of this remains valid even in the context of video services. Therefore, it is relevant to take a brief look at some of this work as well.

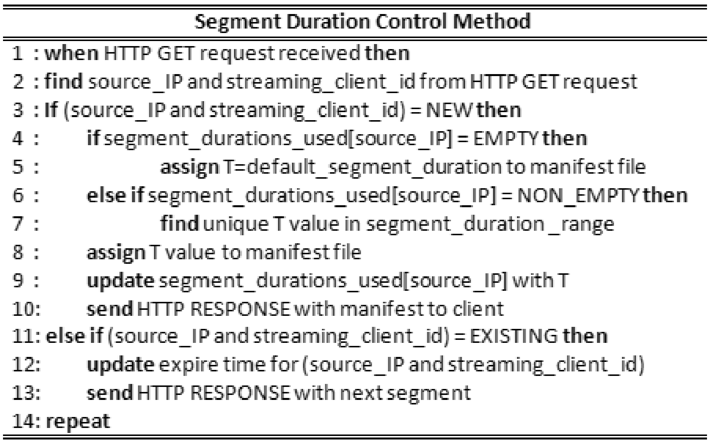
There are several approaches for estimating bandwidth along a network path. As described in [7] there are methods for estimating per-hop capacity, end-to-end capacity, end-to-end available bandwidth and also transport layer capacity and throughput. From an adaptive application point of view, the transport layer approach may seem as most appropriate but the tools in this domain are more of the benchmarking type, rather than real-time tools. Further on, when the adaptive application is TCP based, the work in [8] has shown that any use of active probing should be done with great care in order to avoid reduced TCP throughput due to the actual probing. The published methods for estimating available bandwidth fall into either the Probe Rate Model (PRM) or Probe Gap Model (PGM) categories [9]. The PRM approach is based on the principle of self-induced congestion and by this detecting available capacity, while the PGM approach uses observed inter-arrival time variations for probe packets to estimate the current level of cross traffic, which then combined with knowledge about the total capacity, can be used to produce an estimate of the current available capacity. Common for all methods seen, independent of category - is the use of observed inter-arrival time for probe packets. Intuitively, it would then be good to reduce the range of possible inter-arrival times in order for an accurate bandwidth estimation to take place. This is reflected in many of the methods in this domain, as they quite often work best [10] under the fluid flow traffic assumption. Ignoring the burstiness of traffic has been stated in [11] as one of the important pitfalls when estimating available bandwidth. The challenge of performing accurate bandwidth estimation when bursty traffic is present has also been analytically verified in [12] and stated as valid for most existing methods.

While many publications focus on how to make the client side of an adaptive streaming session more intelligent, the authors of [13] has taken to opposite approach and suggested to make the server side of it more intelligent. Their approach is intended to maintain a low complexity in the player, while at the same time achieving a high degree of effectiveness in session control. Alternative approaches are described in [14] and [15] where the role of the home gateway is investigated in terms of ability to improve adaptive streaming performance.

3. Traffic Shaping Method

The effect of using different segment durations for adaptive video streams delivered to the same user group was suggested and analyzed by means of simulations in our earlier work [16] as a candidate method of improving perceived fairness and stability. Following this, the effect on session fairness was verified by means of measurements in our work [17]. The interesting findings in our earlier work, has provided motivation for also studying the traffic shaping effect of the method. The rationale behind why it could have such an effect is based on the likely higher interleaving degree of traffic to different clients when making them operate with different segment durations. These intervals are equal to the Segment@Quality intervals (cf. Fig. 1), which can be controlled on a per stream basis by the server side through the manifest files. The server side is quite often represented by a CDN (Content Delivery Network) installation inside the same network as the user is directly connected to.

In order for the algorithm to operate it requires the requested video streams to be available with different segment durations, which then corresponds to the time intervals between successive HTTP GET messages from the client side. The list of available segment durations for a requested stream is represented in the pseudocode in Fig. 2 by the segment_duration_range variable.



```
Segment Duration Control Method
when HTTP GET request received then
find source_IP and streaming_client_id from HTTP GET request
If (source_IP and streaming_client_id) = NEW then
    if segment_durations_used[source_IP] = EMPTY then
        assign T=default_segment_duration to manifest file
    else if segment_durations_used[source_IP] = NON_EMPTY then
        find unique T value in segment_duration _range
    assign T value to manifest file
    update segment_durations_used[source_IP] with T
    send HTTP RESPONSE with manifest to client
else if (source_IP and streaming_client_id) = EXISTING then
    update expire time for (source_IP and streaming_client_id)
    send HTTP RESPONSE with next segment
repeat
```

Figure 2. Pseudocode for traffic shaping method

When a HTTP GET request is received from a client it can either be the start of a new video stream or just the request for the next segment of a currently running stream. The algorithm differentiates between these two by the use of the client source IP address (source_IP) and a per client unique stream identification (streaming_client_id). The actual implementation of the latter depends on the specific adaptive streaming solution used, but in the MS Smooth Streaming solution [3] this id is present in all HTTP GET messages. Using this information the server can maintain state information for user groups (i.e. home networks) and apply the segment duration control method.

4. Measurement Setup

In order to verify the effect of the method as described in Section III a testbed was established (cf. Fig. 3) which included a number of clients, a streaming server, a router with bandwidth control and an appropriate switch with port monitoring capabilities connecting a tcpdump probe. The parameters to be studied by means of measurements were packet inter-arrival time (IAT) and video segment fetch duration (SFD), at different congestion levels when the segment duration control algorithm was enabled and not enabled.

An increase in IAT would represent a shaping effect contributing to reduced burstiness in traffic at the lowest level, while an increase in video segment fetch duration would represent a shaping effect on a higher level.

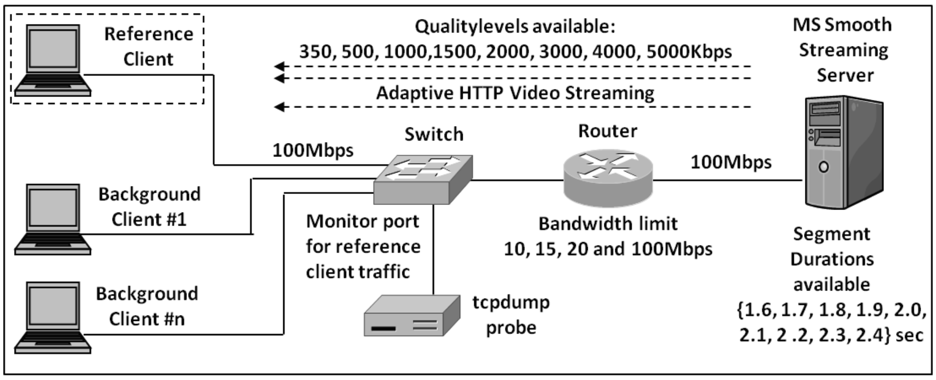

Reference Client
Qualitylevels available:
350, 500, 1000,1500, 2000, 3000, 4000, 5000Kbps
Adaptive HTTP Video Streaming
MS Smooth Streaming Server
Switch
Router
100Mbps
100Mbps
Background Client #1
Monitor port for reference client traffic
Bandwidth limit 10, 15, 20 and 100Mbps
Segment Durations available {1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2 .2, 2.3, 2.4} sec
Background Client #n
tcpdump probe

Figure 3. Measurement testbed

The bandwidth limits used in the experiments were 10/15/20/100 Mbps, configured on the router at IP level. This type of bandwidth limitations allows traffic bursts of configurable size. These burst will appear at wirerate which in this case is 100Mbps. This is important to understand as it gives the required accuracy for the measurement tools to be used. If the server would transmit IP packets back-to-back with sizes between 100-1500 bytes it would require the measurement probe to handle IAT values between 8-120µs. This requirement is within the capabilities of tcpdump as long as the ethernet NIC used is configured correctly in terms of how often it is allowed to deliver the captured packets to the OS kernel for processing.

The traffic measurements were done for the reference client which was always using a default segment duration value of 2 sec, while the background clients was either also using the default value, or a unique value from the available range on the server side. The number of simultaneous adaptive streams used was between 2-5, as per the measurement scenarios described in Table I and II.

Table I. Scenarios for 2 streams

Ref. Client	Bkgd. Client #1	BW limit	Scenario
2.0s	2.0s	{10,15,20}Mbps	EqualT
2.0s	{1.6, 1.7, .. , 2.3, 2.4}s	{10,15,20}Mbps	UniqueT (a-h)

Table II. Scenarios for 3 streams

Ref. Client	Bkgd. Client #1	Bkgd. Client #2	BW limit	Scenario
2.0s	2.0s	2.0s	{10,15,20}Mbps	EqualT
2.0s	{1.6, 1.7, 1.8, 1.9}s	{2.1, 2.2, 2.3, 2.4}s	{10,15,20}Mbps	UniqueT

Table III. Scenario for 5 streams

Ref. Client	Bkgd. Cl. #1	Bkgd. Cl. #2	Bkgd. Cl. #3	Bkgd. Cl. #4	BW limit	Scenario
2.0s	2.0s	2.0s	2.0s	2.0s	100Mbps	EqualT
2.0s	{1.6,1.6, 2.1}s	{1.8,1.7 , 2.2}s	{2.2,1.8 , 2.3}s	{2.4,1.9 ,2.4}s	100Mbps	UniqT (a-c)

All clients are accessing the same adaptive video stream on the server side, with start and stop events synchronized. One measurement period for a specific scenario corresponds to 10 minutes of video streaming. In order to obtain sufficient statistical validity of the results, 30 measurement periods are collected for each scenario. In terms of total measurement time this represents about 242 hours.

5. Results

As an introduction to the measurement results we first present the relevant metrics (IAT and SFD) for a single adaptive video stream, when using the default segment duration value of 2 sec under the different bandwidth scenarios, i.e. 10/15/20/100Mbps.

For the IAT measurement results, these are presented using box and whisker (quantile) plots which give a good view on the key distributional properties location (here: median) and spread (mid 50% values). In addition, we have also included Cumulative Distribution Function (CDF) plots as they provide supplementary information on the location for the observations.

When presenting the results for video SFD observations, we found that distribution plots is appropriate as they give a much clearer view on whether the observed distributions have multiple modes. The importance of investigating this is based on our previous work [16] which gave insight into how unique T periods for competing streams would lead to a variable degree of overlap / no-overlap of the burst periods for each stream. Multi modal characteristics of a distribution can also be seen by CDF plots, but not as clear as in a distribution plot.

5.1 Single session

As seen from the CDF plot in Fig. 4 about 60% of the IAT observations are below 140µs, and 90% are below 200µs. This gives a strong indication of the location in the observed distribution. We also see that there is not much difference between the different bandwidth scenarios, except for a slight gap between the curves in the region between 140-180µs.

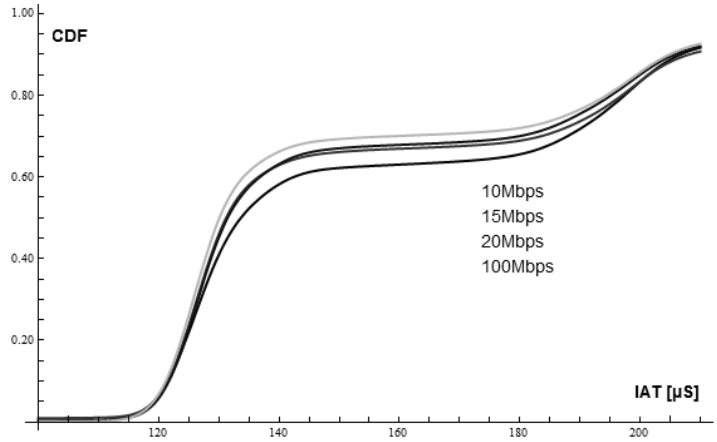


Figure 4. CDF plot for IAT at 10/15/20/100M and single stream

The differences between the different bandwidth levels are marginal in terms of location, but in terms of spread the 10Mbps case stands out with a somewhat wider 50% mid value range (cf. Fig. 5).

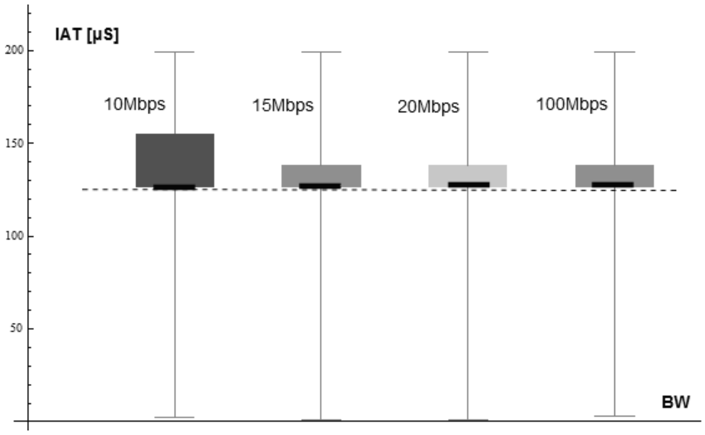


Figure 5. Quantile plot for IAT at 10/15/20/100M and single streams

In general, the similarities make sense since the configured bandwidth settings on IP level in the router allows for bursts up to a certain limit. Experiments using different burst limits from 128KB to 1MB have been performed, but findings are similar to those presented in the following. Only difference is that the distributions are shifted upwards in time scale for smaller burst settings.

The observed wider spread for the 10Mbps case can be explained by the likely case of that the server side tries to maintain a too high average rate over some time scale during the video segment fetch periods.

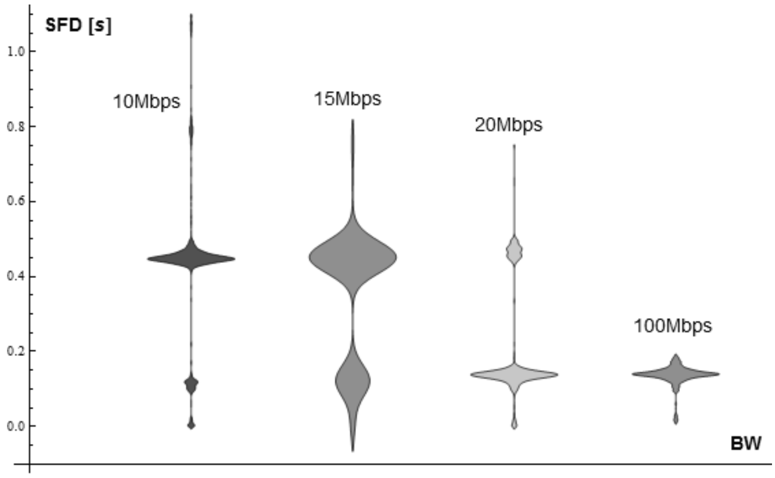


Figure 6. Distribution plot Video SFD at 10/15/20/100M, single stream

When we look at distributions for the video SFD observations (cf. Fig. 6) we see that for the two lower bandwidth levels the locations are around 0.5seconds, while for the two highest bandwidth levels around 0.2 seconds. The latter is then likely to be the server side maximum. Knowing that the single stream observed was able to operate at the highest quality level (5Mbps) in all bandwidth cases and that a T value of 2 sec was used, it means that the server side tries to deliver each video segment at rates between 20-50Mbps.

This initial characterization of the traffic generated from adaptive video streaming clearly shows that the traffic pattern is very bursty both in terms of packet IAT and also video SFD. Whenever the server side gets a request for a new video segment, it will send the corresponding data segment back at wirespeed (in this case 100Mbps), only limited by TCP mechanisms.

5.2 Two simultaneous sessions

When two sessions of the type earlier described are competing for a bandwidth of 15Mbps, both sessions will achieve their maximum quality level, i.e. 5Mbps. Thus, for this specific scenario - the video segment sizes (in bytes) requested by each HTTP GET message during the session lifetime will be equal. The reason why we focus on the results for this type of non-congested scenario is that it allows for a simultaneous study of both packet IAT and video SFD.

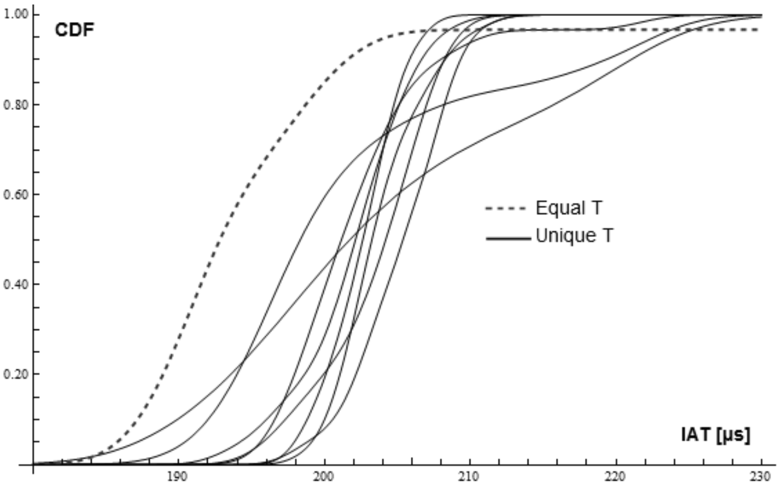


Figure 7. CDF plot for IAT at 15M and 2 streams

The CDF plot for IAT measurements Fig. 7 show a clear difference between the cases where both sessions are using equal T and where they use unique T (cf. Table I) values. It can be seen that the unique T method leads to that around 70% of the lowest IAT observations are shifted upwards.

When the same IAT measurements are viewed by means of quantile plots (cf. Fig. 8) we also see the shift in location represented by the mid 50% values and also a higher median value for all unique T scenarios (labeled a-h). The difference in location (median) between the equal T case and the unique T cases is in average 10.2 µs.

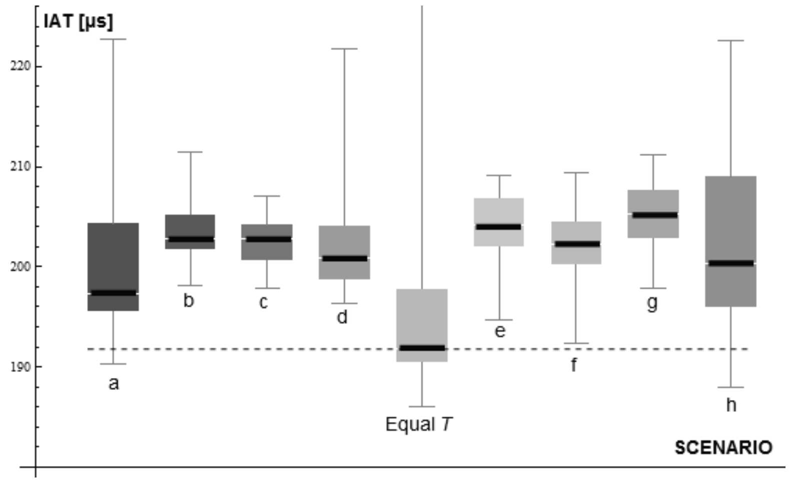


Figure 8. Quantile plot for IAT at 15M and 2 streams

The unique T cases a/h stand out with a higher spread (cf. Fig. 8). These are the cases where the differences between segment durations used by the two competing sessions are largest, i.e. 0.4sec (cf. Table I).

From an IAT perspective the results are similar also at the 10Mbps and 20Mbps scenarios, with an average difference in location at respectively 7.1µs and 10.6 µs (cf. Table IV). The high spread for the unique T cases a, and h is also present here.

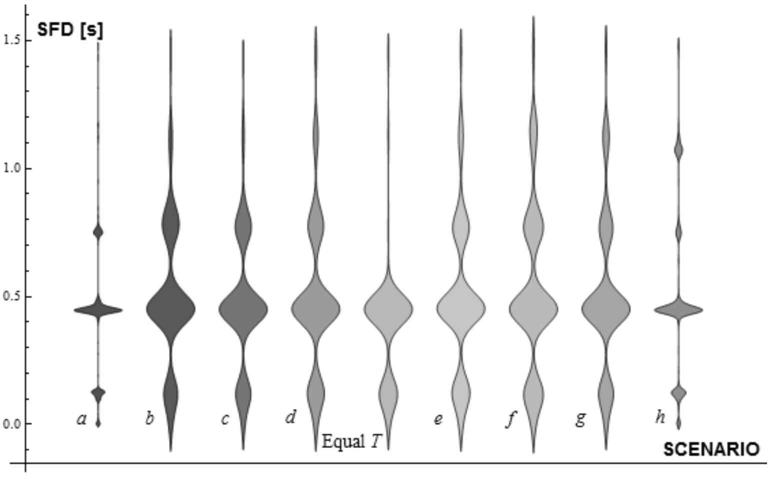


Figure 9. Distribution plot Video SFD at 15M and 2 streams

The measurement results for video SFD in Fig. 9 show an interesting effect in terms of the distribution modes. For all unique T cases a new mode of the distributions has appeared which is located around 0.75seconds, which contains about 30% of the SFD observations. This change represents a shaping effect as it indicates longer burst periods.

5.3 Three simultaneous sessions

Also in the case of three simultaneous sessions, and a bandwidth of 20Mbps – all sessions will achieve their maximum quality level. Thus we have a scenario with the same characteristics as in the previous.

The IAT measurement results are given in Fig. 10 and Fig. 11, and they both show a clear difference between the cases where both sessions are using equal T and where they use unique T (cf. Table I) values. In this case, we see that the unique T method leads to about 40% of the lowest IAT observations being shifted upwards.

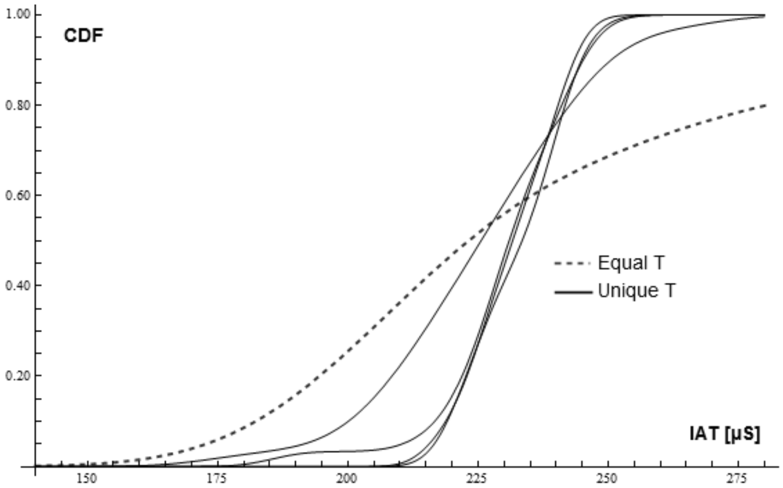


Figure 10. CDF plot for IAT at 20M and 3 streams

The difference in location (median) between the equal T case and the unique T cases (cf. Fig. 11) is in average 11.8 µs and the reduced spread in the mid 50% values is quite clear.

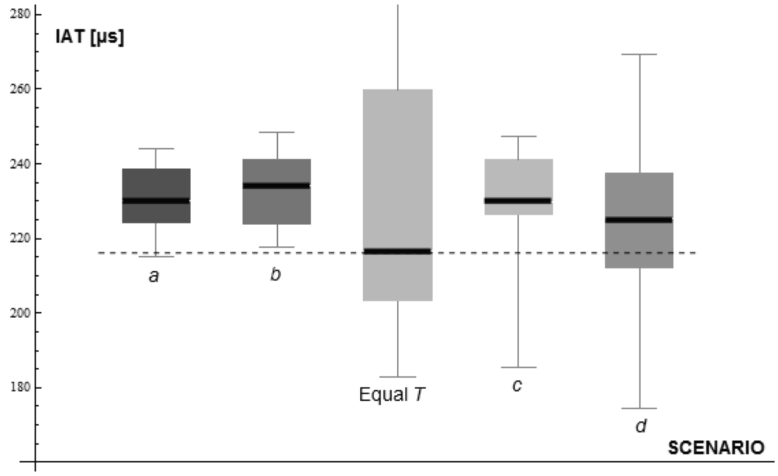


Figure 11. Quantile plot for IAT at 20M and 3 streams

From an IAT perspective the results are similar also at the 15Mbps scenarios, with an average difference in location at 12.5 µs (cf. Table IV). The high spread for the unique T cases a, and h is also present here. Regarding the 10Mbps scenario the findings are different, but keeping in mind the high degree of congestion in that case it may be an indication of that there is a limit for how much congestion there can be for the shaping method investigated in this paper to have effect.

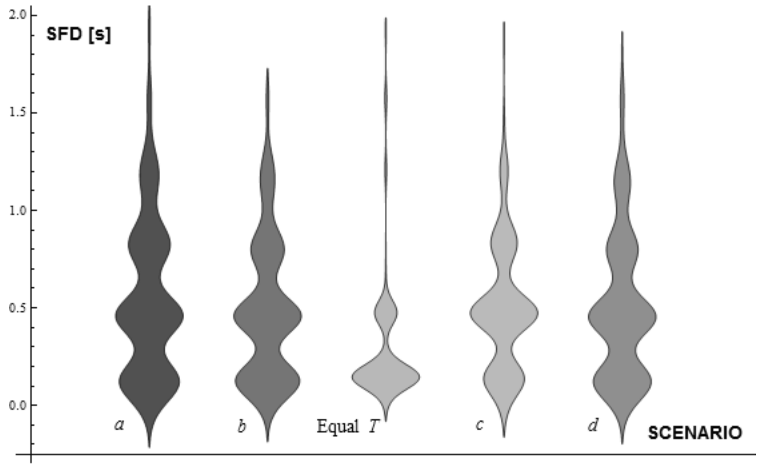


Figure 12. Distribution plot Video SFD at 20M and 3 streams

The measurement results for video SFD in Fig. 12 show the same effect as in Fig. 9 in terms of new modes in the distributions appearing. In this case, it seems as if two new modes appear and they are both located in the higher regions above 0.6sec. The new modes contain about 35% of the SFD observations. Again, this change represents a shaping effect as it indicates longer burst periods.

5.4 Five simultaneous sessions

The final experiment was done at 100Mbps access speed, i.e. no limitation on IP level. At this bandwidth level and the maximum quality level for each stream still at 5Mbps, all clients are able to reach their highest level.

The reference session is still using the segment duration of 2sec, while the other four are using values as per Table III.

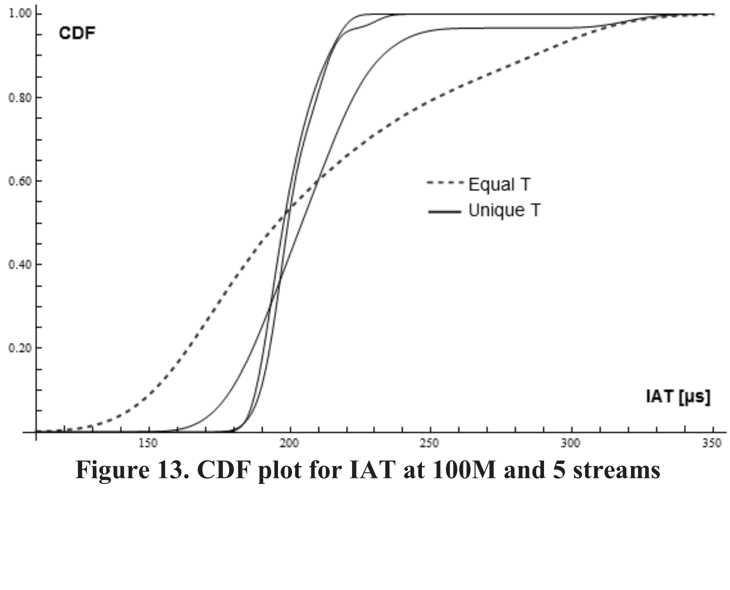


Figure 13. CDF plot for IAT at 100M and 5 streams

The findings even for this scenario are in line with the results in the previous sections as the location of the unique T distribution is shifted upwards (cf. Fig. 13). Also here, the amount of IAT observations being shifted upwards is about 40%.

Further on, the quantile plot in Fig. 14 show that the mid 50% values are shifted upwards and spread is reduced. The increase in median value is also noticeable.

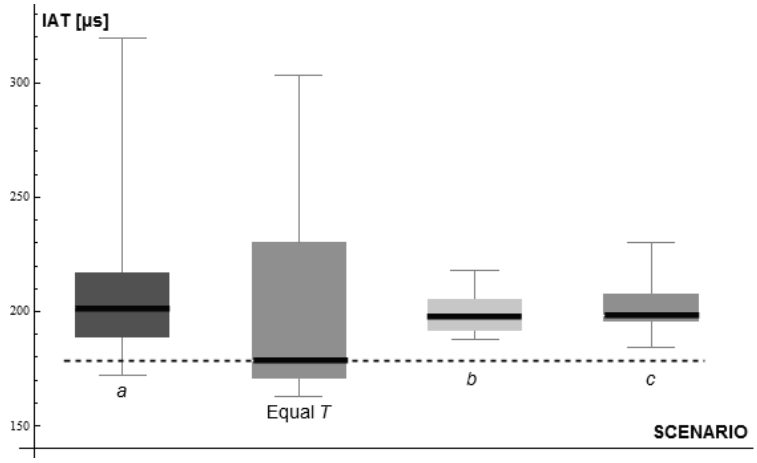


Figure 14. Quantile plot for IAT at 100M and 5 streams

The measurement results for video SFD in Fig. 15 does not indicate the same effect in terms of new modes appearing in higher regions, which was the case for in results presented for 15Mbps/2 sessions and 20Mbps/3 sessions. Instead, we notice a shift downwards in location for the distributions. Therefore, we cannot state a shaping effect at this level for video SFD.

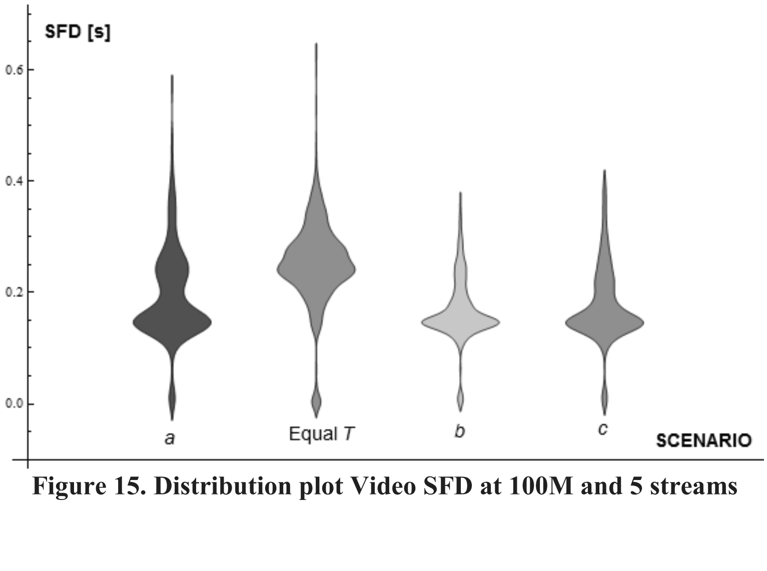


Figure 15. Distribution plot Video SFD at 100M and 5 streams

6. Summary

The effects of applying the segment control method on competing streams delivered to the same network destination, in terms of change in location and spread for the observed IAT distribution are summarized in Table IV, V and VI.

Table IV. Scenarios for 2 streams

	ΔLocation (µs)			**ΔSpread (µs)**		
	Min	Avg	Max	Min	Avg	Max
10Mbps	1.3	7.1	12.0	0.4	-4.6	-7.6
15Mbps	5.7	10.2	13.4	5.6	-1.4	-3.9
20Mbps	4.0	10.6	13.6	4.2	-4.8	-8.7

Table V. Scenarios for 3 streams

	ΔLocation (µs)			**ΔSpread (µs)**		
	Min	Avg	Max	Min	Avg	Max
10Mbps	-20.1	-8.8	7.4	-17.9	-25.8	-34.9
15Mbps	9.6	12.5	16.3	-14.7	-19.5	-24.9
20Mbps	6.9	11.8	16.3	-30.9	-38.5	-41.2

Table VI. scenarios for 5 streams

	ΔLocation (µs)			**ΔSpread (µs)**		
	Min	Avg	Max	Min	Avg	Max
100Mbps	18.7	20.6	23.4	-31.4	-41.7	-47.6

The ΔLocation column shows the change in location (median) for the observed IAT distribution and the ΔSpread column shows the change in width of the mid 50% quartile. For both metrics - minimum, average and maximum values are given. A positive value for ΔLocation represents an increase in IAT and a negative value for ΔSpread represents a reduced spread of the IAT observations. As such, positive values for ΔLocation should be interpreted as good while for ΔSpread, a negative value is good. The latter holds only if positive values for ΔLocation are observed at the same time.

The findings for the effect on video SFD observations when applying the unique T method are positive for all bandwidth scenarios studied, except for the 100Mbps case. The deviating results seen for the latter case does not necessarily mean that the same effect is not present, as it could be that we just need to increase the measurement period durations in order to capture the effect. The need for longer measurement for high bandwidth cases are to some extent justified in the analytical parts of our previous work [16].

It should be noted that video SFD has only been measured for cases when streams were allowed to operate at their maximum level, and this was done in order to have constant segment size (in terms of bytes). Studying the shaping effect at SFD level in cases where the stream quality levels changes a lot is a somewhat more challenging task which was outside the scope of this paper.

7. Conclusions

The results show that the segment duration control method has an effect on the measured traffic to the reference client in terms of an upwards shifted location and also reduced spread in the IAT observation distribution. This means that the method contributes to an increase in packet IAT and also that the variations for this metric is reduced. The increase in packet IAT represents a traffic shaping effect, and therefore the method can be stated to have such an effect for all studied scenarios, except for the one case where the congestion is very high (10Mbps, 3 sessions). The shaping effect contributes to a reduced burstiness in the traffic and therefore has a potential of making available bandwidth estimation more accurate.

The reduced spread in IAT observations can also have a positive effect on available bandwidth estimations as most methods in this domain relate to observed packet delay for the injected probe traffic. By making the range of experienced delay smaller, it is reasonable to believe that the accuracy will be improved.

Further on, we consider it very important that we can see a simultaneous positive effect on both packet IAT and video segment fetch duration (SFD). This strengthens our documentation of the shaping effect by using our suggested [16] [17] and investigated segment size control method.

Whether the impact of the segment duration control method is significant enough for it to have a real effect depends on the available bandwidth method used, and its real-time capabilities. Our earlier work [17] provides some justification in this regard, as it shows that the method contributes to improved fairness among competing sessions. However, it is not proven that this is directly caused by the traffic shaping effect documented in this paper.

Therefore, at this time - we can only state that the suggested method does provide a shaping effect at both packet IAT and video SFD level. The significance of these effects for applications such as Adaptive HTTP Video Streaming should be closer studied.

8. Future work

As future work in this domain it would be interesting to analyze the accuracy of available real-time available bandwidth algorithms for applications such as Adaptive HTTP Video Streaming. It would then be of great interest to seek an indicator of how much shaping or reduced delay variation is needed in order for the algorithms to become even more accurate. It would also be interesting to use the obtained knowledge about the traffic patterns for adaptive video streaming and develop enhanced available bandwidth algorithm especially addressing this service type. A combination of using information from segment data transferred in busy periods and some active probing in idle periods may be a candidate approach. Combining this with information from the home gateway is also of interest.

9. Acknowledgment

The reported work is done as part of the Road to media-aware user-Dependant self-aDaptive NETWORKS - R2D2 project. This project is funded by The Research Council of Norway.

References

[1] I. Sodagar, “The MPEG-DASH standard for multimedia streaming over the internet,” Multimedia, IEEE, vol. 18, no. 4, pp. 62 –67, April 2011.

[2] H. Riiser, H. S. Bergsaker, P. Vigmostad, P. Halvorsen, and C. Griwodz, “A comparison of quality scheduling in commercial adaptive http streaming solutions on a 3G network,” in Proceedings of the 4th Workshop on Mobile Video, MoVid ’12. New York, USA: ACM, 2012, pp. 25–30.

[3] A. Zambelli, “IIS Smooth Streaming technical overview, Tech. Rep from Microsoft, March 2009.

[4] C. Liu, I. Bouazizi, and M. Gabbouj, “Rate adaptation for adaptive http streaming,” in Proceedings of MMSys 2011, pp. 169-174.

[5] L. Chenghao, I. Bouazizi, and M. Gabbouj, “Segment duration for rate adaptation of adaptive http streaming,” in Multimedia and Expo (ICME), 2011 IEEE International Conference on, july 2011, pp. 1 –4.

[6] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, “Qdash: a QoE-aware dash system,” in Proceedings of MMSys 2012, pp. 11–22.

[7] C. Dovrolis, R. Prasad, M. Murray, and k. claffy, “Bandwidth estimation: metrics, measurement techniques, and tools,” IEEE Network, vol. 17, no. 6, pp. 27–35, Apr 2003.

[8] A. Johnsson and M. Björkman, “Measuring the impact of active probing on tcp,” in International Symposium on Performance Evaluation of Computer and Telecommunication Systems, July 2006.

[9] J. Strauss, D. Katabi, and F. Kaashoek, “A measurement study of available bandwidth estimation tools,” in Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, ser. IMC ’03. New York, USA: ACM, 2003, pp. 39–44.

[10] X. Liu, K. Ravindran, B. Liu, and D. Loguinov, “Single-hop probing asymptotics in available bandwidth estimation: sample-path analysis,” in Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, ser. IMC ’04. New York, USA: ACM, 2004, pp. 300–313.

[11] M. Jain and C. Dovrolis, “Ten fallacies and pitfalls on end-to-end available bandwidth estimation,” in Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, ser. IMC ’04. New York, USA: ACM, 2004, pp. 272–277.

[12] X. Liu, K. Ravindran, and D. Loguinov, “A stochastic foundation of available bandwidth estimation: multi-hop analysis,” IEEE/ACM Trans. Netw., vol. 16, no. 1, pp. 130–143, Feb. 2008.

[13] L. De Cicco, S. Mascolo, and V. Palmisano, “Feedback control for adaptive live video streaming,” in Proceedings MMSys 2011, pp. 145–156.

[14] R. Houdaille and S. Gouache, “Shaping http adaptive streams for a better user experience,” in Proceedings of MMSys, 2012, pp. 1–9.

[15] B. Villa and P. Heegaard, “Towards knowledge-driven QoE optimization in home gateways,” in Proceedings of ICNS 2011, May 2011.

[16] B. Villa and P. Heegaard, “Improving perceived fairness and QoE for adaptive video streams” in Proceedings of ICNS 2012, March 2012.

[17] B. Villa and P. Heegaard, “Improving fairness for adaptive http video streaming,” EUNICE 2012, August 2012

PAPER 8: Detecting Period and Burst Durations in Video Streaming by means of Active Probing

Bjørn J. Villa, Poul E. Heegaard

Presented at
The 3rd International Conference on Computer Communication and Management (ICCCM 2013)
May. 19-20, 2013, Copenhagen, Denmark

Published in
International Journal of Computer and Communication Engineering
ISSN 2010-3743, volume 2 (4), 2013, pp. 460-467

Detecting Period and Burst Durations in Video Streaming by means of Active Probing

Bjørn J. Villa, Poul E. Heegaard

Abstract: This paper presents a method for detecting periodic behaviour and burst durations in IP traffic observed on access links. Periodic behaviour in traffic observed on the Internet is growing due to the increasing amount of services with video components. The method is based on active probing and serial correlation of the observed probe packet inter-arrival times on the receiver side. Verification of the method is done based on an experimental implementation, using both controlled lab services and live public services. The services used in the experiments are adaptive video streaming based on Microsoft Smooth Streaming and Netflix. The results show that the method provides accurate results for both the estimator of service period and the burst durations. The findings are similar both for the controlled lab experiments and the live network measurements. The amount of probe traffic required in order for the method to perform satisfactory is very moderate.

Keywords: Periodic Traffic Patterns, Traffic Burst Duration, Active Probing, Serial Correlations, Adaptive Video Streaming, Access Links.

1. Introduction

The amount of services provided to Internet users around the world following an Over-The-Top service delivery model is increasing. This model is based on that the involved network operators are not taking any active measures in order to assure the required Quality of Service (QoS) levels of the specific services. Traffic is carried as part of the best-effort class and will therefore face obvious challenges in terms of being able to meet the end users expectations regarding Quality of Experience (QoE).

The best-effort traffic class on the Internet is a rough place, and the handling of traffic is unpredictable. This can lead to fluctuations in the experienced QoS metrics with the potential of a negative service effect. The engineering approach to this with an Over-The-Top perspective is to make the services adaptive. The type of adaption required is for the service itself to be able to adjust their QoS requirements. The case of fixed QoS requirements, e.g. with regard to bandwidth or even packet loss ratio would not work well in a best-effort network class.

The strong growth of services with video components over the last years, are predicted to become even stronger in the years to come [1]. This success is partly caused by the emerging solutions and standard [2] for adaptive video streaming. The adaptive nature of these services have proven to be very effective, and enabled new business models (e.g. Netflix, YouTube) on the Internet for content which earlier was considered impossible to deliver in a best-effort QoS class.

An interesting consequence of the growing amount of video services, and adaptive ones in particular – is that the traffic patterns on the Internet are changing. This alone is of course interesting enough, but in the context of adaptive services it may also create some interesting effects. An interesting research question arises from this, as it will become more and more common that adaptive services must adapt according to other adaptive services. Whether this will lead to a continued or reduced success in terms of effectiveness for this type of services remains to see.

1.1 Problem Statement

The research topic addressed in this paper covers a method to detect and characterize services with a periodic behavior on access links, by means of active probing. As basis for this work we have used adaptive video services, which typically embed this behavior. Thus, the method does not address the specific recognition of a periodic service as being video or something else. The method focuses on detecting the period (if any) and also the burst duration of traffic inside each period. Further on, a basic evaluation of how much probe traffic is needed and also the required sample size is covered.

The investigated method provides estimates for parameters (T_p, T_b) indicated in Fig. 1. The periodic pattern of the video service used in the experiments consists of burst and idle periods. However, it should be noted that even in the idle period there is traffic, but much less than in the burst periods.

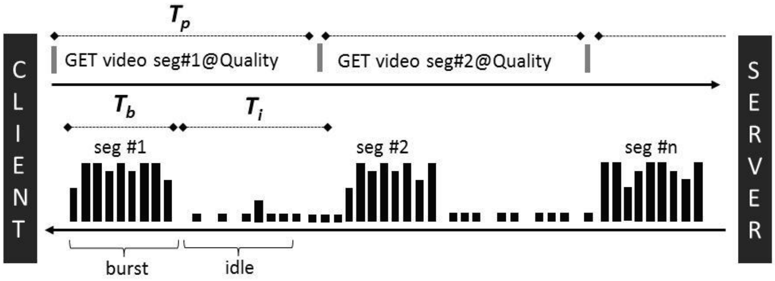


Figure 1. Traffic Pattern for a Video Service

The value of this work can be viewed as both a general contribution to the discipline of service characterization, but also as a candidate component of future methods for estimating available bandwidth. Obtaining knowledge of periodic components in the cross-traffic and also the duration of burst periods has a potential of increasing effectiveness and accuracy in available bandwidth estimations.

1.2 Research Approach

The chosen approach for investigating the suggested method of detecting periodic behavior and burst duration, is by means of experiments in a controlled lab environment and a real network with public services. In the lab environment we use an adaptive video streaming service based on the Smooth Streaming platform from Microsoft, while in the live network we are accessing a the public Netflix service. The required background load in the lab environment is provided by means of a CBR type traffic source. The complete measurement setup will be further described in Section V.

1.3 Paper Outline

The structure of this paper is as follows. Section II provides an overview of related work; Section III provides the relevant characteristics of adaptive video streaming as captured by passive measurements; Section IV describes our method of detecting periodic traffic patterns and burst duration by means of active probing; Section V describes our hybrid passive and active measurement setup; Section VI presents the results based on the active probing and compares them with the passive measurements; Section VII presents our conclusions and an outline of future work is given in Section VIII.

2. Related Work

The study of phenomena's observed in Internet service usage and traffic patterns generated are done in different ways. Quite often the phenomenon of interest is required to be studied over time in order to obtain sufficient information. In such cases, the tools and methods available for analyzing a discrete set of time-ordered data (i.e. time series) are quite useful [3]. For the purpose of detecting periodic components in traffic pattern and the duration of such, estimators for serial correlation [4] [5] (also known as autocorrelation) are known to be very efficient.

In the area of available bandwidth estimations along a network path, there are many different approaches to how this can be done using active probing techniques [6] [7]. The active probing results in a time series of observations, such as changes in delay components [8]. These observations are then used as input to the respective algorithms for estimating the available bandwidth. There are many challenges of performing such estimations in an accurate manner [9], among which neglecting the burstiness of cross-traffic is one. Among the more recent methods in this domain, the approach described in [10] is quite interesting. It aims at estimating available bandwidth in real-time and does this by applying a filter-based method. The idea of using filtering as part of the continuous processing of collected observations in our own work is inspired by this.

Especially on access links, the traffic patterns observed are dominated by burst components. The reason for this is composed by different factors. First of all, the strong dominance of TCP based applications [11] has been shown to generate traffic bursts in short time scales [12]. This follows directly from the TCP protocol behavior in terms of timeout events and congestion avoidance. Another important factor which leads to bursty traffic is the strong dominance [1] [13] of services with video components. In our previous work [14] we studied the packet inter-arrival time distributions for this type of services, and suggested a new method to achieve a shaping effect. The typical traffic pattern observed when video services are present has the signature of periodic components, containing a number of bursts.

With regard to the correlation structures in a time series of observed delay components, it has been shown [15] that the TCP protocol tends to generate traffic patterns with this property in sub-second time scales. In addition, the nature of services used also contributes to correlation structures in the time scale of seconds. The authors of [16] showed that they were able to detect Skype traffic by investigating correlation between traffic bursts in network traffic traces. A similar effect was shown in [17] where the focus was on video services.

A more general study of how correlation structures in traffic patterns can be obtained either through sampling or active probing is presented in [18]. In this, estimators for correlations of network traffic are described and experimental results presented. The estimators are applicable for both samples based on passive and active monitoring.

In our earlier work [19] [20], we studied the behavior and characteristics of adaptive video streaming from different perspectives. Further on, in our work in press [14] we presented a new method for achieving a traffic shaping effect for adaptive video streaming, without involvement from network components.

3. Characteristics of video streaming services

The periodic nature of an adaptive video streaming service is given by the repeated requests for the next segment in a video stream, at a specific quality level.

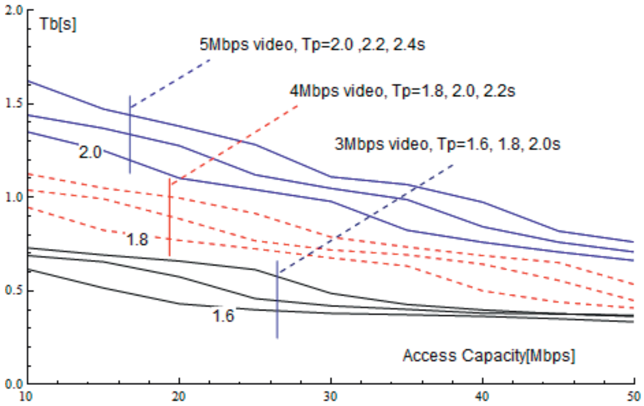


Figure 2. Passive measured burst durations for MS Smooth Streaming

The passive measured burst periods for quality levels at 3/4/5Mbps over the access capacity range from 10-50Mbps are shown in Fig. 2. The segment request interval (T_p)

used in the experiments for the MS Smooth Streaming [21] based service is between 1.6 and 2.4 sec. The interval granularity available in the specific solution was 0.1s, which gave five different T_p scenarios available for use in the experiments. Only a selection of this is shown in Fig. 2 for the different video stream quality levels and access capacity levels.

For the public Netflix service used in our experiments, we did not have the opportunity to create a similar wide range of scenarios as for the MS Smooth Streaming service. The obvious reason for this would be that it was a live service, and Netflix resources were not involved in our research. The characteristics of the Netflix streaming were studied in depth in [22], and significant differences between Netflix and MS Smooth Streaming solution are described. Relevant to our work are their findings that the Netflix streaming uses two TCP connections and also that the request intervals relates to data volumes rather than duration.

For a specific movie, accessed from Netflix in both their stated HD quality and also as non-HD the passive measured burst periods by using TCPdump (cf. Fig. 8) are presented in Fig. 3.

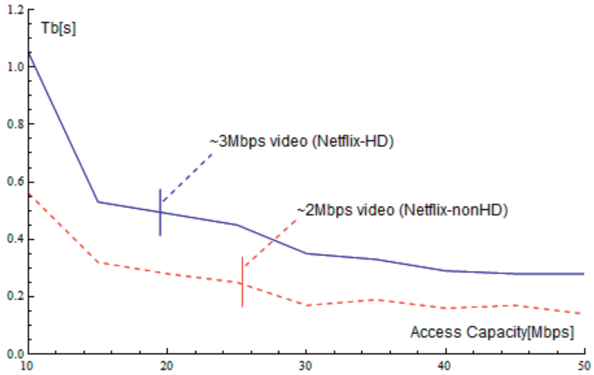


Figure 3. Passive measured burst period durations for Netflix

In order to get a view on the traffic pattern generated by Smooth Streaming and Netflix in a scenario with a minimum of constraints, we measured the average burst periods using 100Mbps access capacity. In Table I the results obtained for all available video quality levels are presented. It should be noted that the Netflix service may be subject to some limitations outside of our control as it resides outside of the lab environment.

Table I. Passive measured Burst Periods at 100Mbps

Burst Period	2Mbps	3Mbps	4Mbps	5Mbps
Smooth Streaming with T_p=2.0s	- na -	0.07s	0.09s	0.12s
Netflix	0.13s	0.21s	- na -	- na -

As a consequence of the Netflix request intervals being decided by data volumes, rather than playback time – the periodic nature of this streaming type could be questioned. The effect of using multiple TCP session is also interest in this regard. In Fig. 4, the estimated probability density functions (pdf) for passively measured time gaps between successive HTTP GET messages, per TCP session and combined are shown. The pdf estimations are based on measurement of the request intervals for the Netflix-HD movie at 30Mbps access.

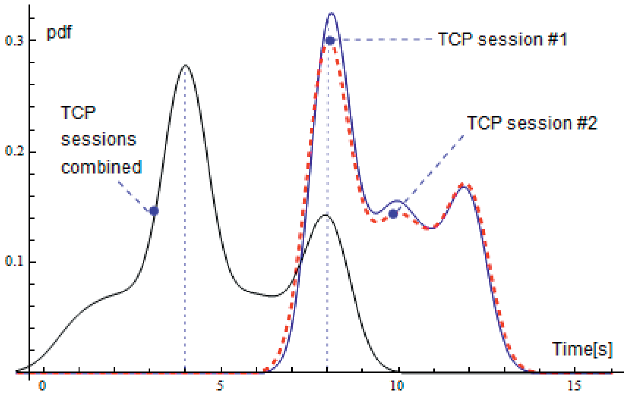


Figure 4. Estimated PDF for Netflix period durations

It is interesting to see that each of the two TCP sessions has a location in their distribution around 8 seconds. They also seem to multimodal, with peaks in additions to the dominating one at 8sec. When the traffic from the two sessions is studied together as they appear on the wire, we get a distinct shift in the distribution. The location for the combined pdf is around 4sec. This indicates that the TCP sessions are interleaved in such a way that a periodic pattern could be expected to have a period lower than each of the TCP sessions.

4. Method

The method used in order to detect and estimate periodic behavior and burst duration of traffic on an access link is by mean of active probing, and a serial correlation at different lags for the times series of processed probe packet inter-arrival times.

4.1 Probe traffic generation

The probe traffic is generated according to trace files with configurable packet size and inter-packet times. The specification of the probe traffic is assumed known to the receiver. Further on, the receiver must be able to tell the difference between a probe packet received and any other traffic, and also that it is able to timestamp the received packets. As illustrated in Fig. 5, the time between probe packets sent is noted t_{in} and the measured inter-arrival times between consecutive probe packets at the receiver side is noted $t_{out,i}$. The time between packet pairs is noted t_{gap}.

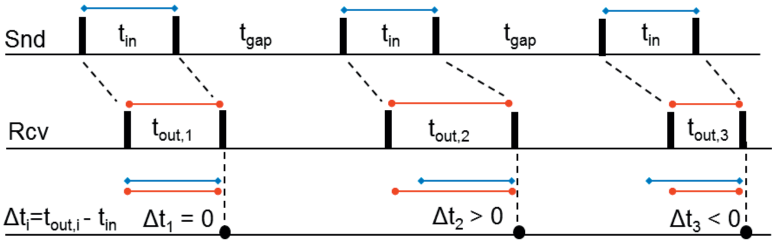


Figure 5. Packet-Pair based probe traffic

Depending on the cross-traffic profile, the receiver may see either an increase or decrease in the probe packet spacing (Δt_i). Recognizing that even cross-traffic during the t_{gap} periods may impact the $t_{out,i}$ samples it leads us towards some kind of processing of the samples before using them further.

4.2 Processing of received probe traffic

In order to simplify the analysis part we found that a basic filtering mechanism for the observed $t_{out,i}$ values was very efficient. The purpose of this is to filter out only those $t_{out,i}$ values with a clear indication of that significant cross-traffic has occurred during the corresponding time interval. The filtering approach is based on using a cumulative moving average CM_i (cf. Eq. 1) of the $t_{out,i}$ samples, rather than the known probe packet spacing t_{in} as basis for deciding whether a sample indicates cross-traffic or not.

$$CM_i = \frac{t_{out,1} + t_{out,2} + \ldots + t_{out,i}}{i} \qquad (1)$$

The time series T_{out} of observed $t_{out,i}$ values is then passed through the filter (cf. Eq. 2) using the calculated CM_i value as input. The resulting time series is noted T_F and contains elements noted $t_{f,i}$.

$$\begin{gathered} \forall\, t_{out,i} \in T_{out} \\ if\ t_{out,i} > CM_i\ then\ \ t_{f,i} = t_{out,i} \\ if\ t_{out,i} \le CM_i\ then\ \ t_{f,i} = CM_i \end{gathered} \qquad (2)$$

The elements of the time series T_F is then used as input to computation of lag-s serial correlation X_s (cf. Eq. 3). For the sake of reducing computation time, and also the time required to detect and estimate period and burst duration, the maximum lag investigated should be set to a reasonable level. The chosen level should match a threshold in time, which reflects the highest potential period of interest to us.

$$x_s = \frac{\sum_{i=1}^{N-s}(t_{f,i} - \overline{t_f})(t_{f,i+s} - \overline{t_f})}{\sum_{i=1}^{N} \left(t_{f,i} - \overline{t_f}\right)^2} \qquad (3)$$

In our case, since the probe packets are equally spaced in time, the lag value s directly maps over to time by simply multiplying it with the probe packet period.

4.3 Analyzing output from serial correlation

The output of lag-s serial correlation of the time series T_F gives a new time series consisting of correlation values X_s. Our method of detecting and estimating period and burst duration in the video streaming services used in the experiments is based on inspection of the X_s values. To demonstrate the strengths of this approach, we would first like to present a graphical view of the serial correlation of a theoretical discrete signal F_i (cf. Fig. 6) with a period of 300.

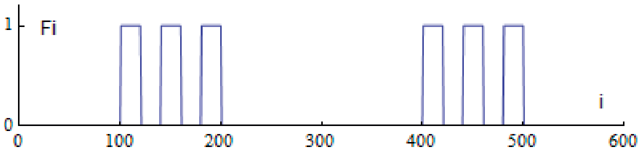


Figure 6. Theoretical discrete periodical signal

This signal can also be represented as a time series consisting of a repeating pattern of length 300 with values of either 0 or 1. When the lag-s serial correlation is computed for this, we get an output as shown in Fig. 7.

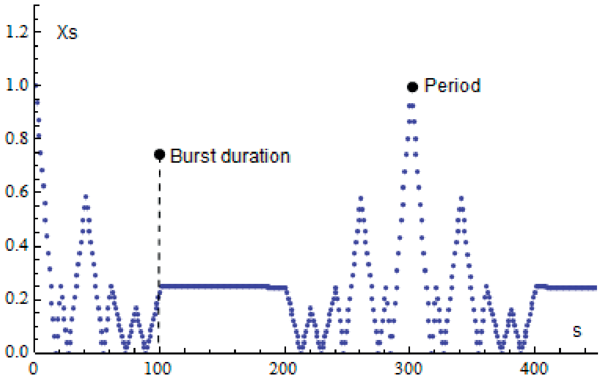


Figure 7. Serial correlation for theoretical signal

From investigating the graphical view of the serial correlation we can clearly see the period of the original signal indicated by the peak correlation value at lag 300. We can also see the width of 100 for the sub-period in the original signal which contains the three spikes. These capabilities of the serial correlation are well described in [5], and we will use this as basis for analyzing our measurement results for the purpose of validating our method.

5. Measurement Setup

The hybrid active and passive measurement setup used for performing the experiments related to this paper contains several components, ranging from the client side over to the server side as shown in Fig. 8. On the client side the video service is accessed by a dedicated Windows based PC, while the client receiving the probe traffic is a separate Linux based PC. On the server side, we have the dedicated Microsoft Smooth Streaming server and also the probe traffic generator For the purpose of access to the live Netflix service, the lab is connected to the Internet. The access network part consists of commercial off-the shelf products which give access to useful functions for controlling bandwidth similar to those used in commercial networks.

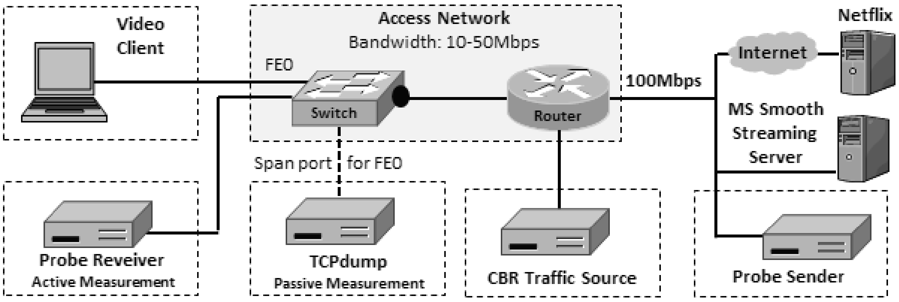


Figure 8. Hybrid active and passive measurement testbed

The CBR traffic source is based on the Click Modular Router [23], which is a simple solution for generating basic types of traffic. The probe traffic sender and receiver which are used for the active measurements are based on the CRUDE/RUDE tool [24], which has the required capabilities of generating traffic pattern based on trace files. The TCPdump node is used to provide the actual characteristics of the services by means of passive measurements, as presented in Section III. These characteristics serve as basis for verifying the effectiveness and accuracy of the method we are studying.

6. Results

The results to be presented in this section cover both experiments with Smooth Streaming in a lab environment, and similar experiments using the live Netflix service. The results for the controlled lab service is more elaborate than the Netflix service. The characteristics for both services captured by means of passive measurements (cf. Section III) should be used as reference point when considering the following results.

In the presentation of serial correlation results for the probe traffic inter-arrival times, the lag parameter has been converted to time for the sake of making the results easier to read. The lag to time conversion follows directly from the probe period used.

6.1 Smooth 5Mbps Streaming on 100Mbps Access

The scenario with potential of having the most burst oriented traffic is the scenario with full access speed (100Mbps) and a single service active using any one of the available T_p values between 1.6s and 2.4s. As we can see from Fig. 9, the computed serial correlations using the method described in Section IV gives very distinct peaks at the expected locations, i.e. at the T_p values being used.

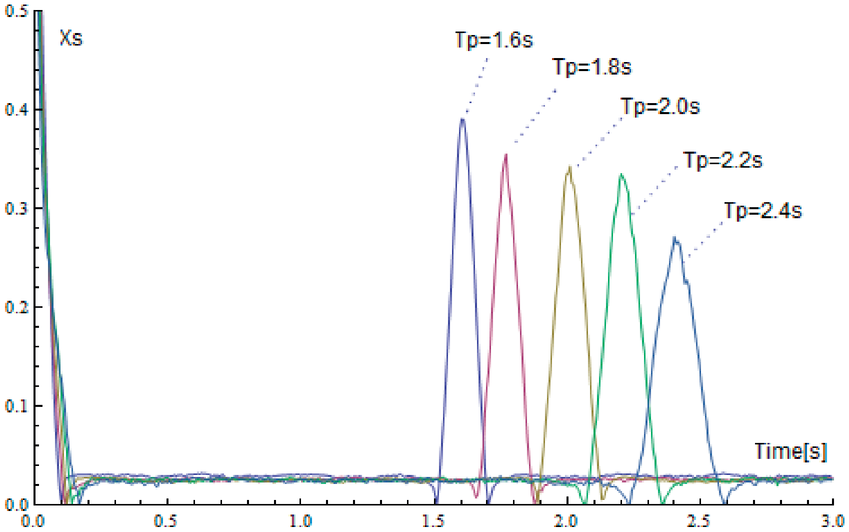


Figure 9. Period detection by active probing

For this scenario, our method for detecting the periodic nature in the video traffic is quite accurate. The peaks are easy to detect both by means of graphical views and by pure computation.

When we focus in on the lower range of the time axis, we also see the presence of serial correlation between probe packet IAT observations within a burst (cf. Fig. 10). The T_b values based on passive measurements from Table I have been included in the illustration, for each of the T_p cases.

We observe that the serial correlation goes to zero at the point where we have reached the burst duration. The match between the passively measured T_b values, and those indicated by the points where the serial correlation reaches zero are very close.

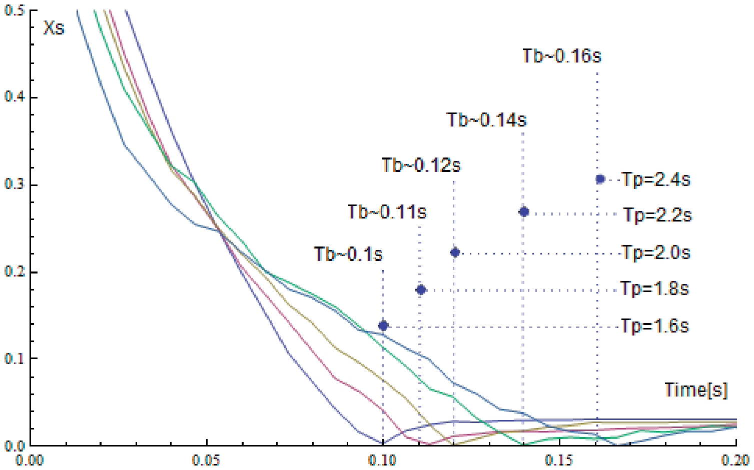


Figure 10. Estimated burst duration by active probing

In this scenario, both the graphical view and a computational approach would be able to present estimators for T_b based on the output from serial correlation X_s.

6.2 Smooth 5Mbps Streaming on 25Mbps Access

In the majority of scenarios investigated, the access capacity represents a limiting factor with regard to potential burst rate for the video stream. These are the cases where the access speed is lower than the interface speed on the video server side. This could be considered as the typical scenario in a real use case.

In Fig. 11, the serial correlation results obtained when using a 25Mbps access are shown. For the sake of clarity, only X_s output when using two different T_p values (2.2 and 2.4) are shown.

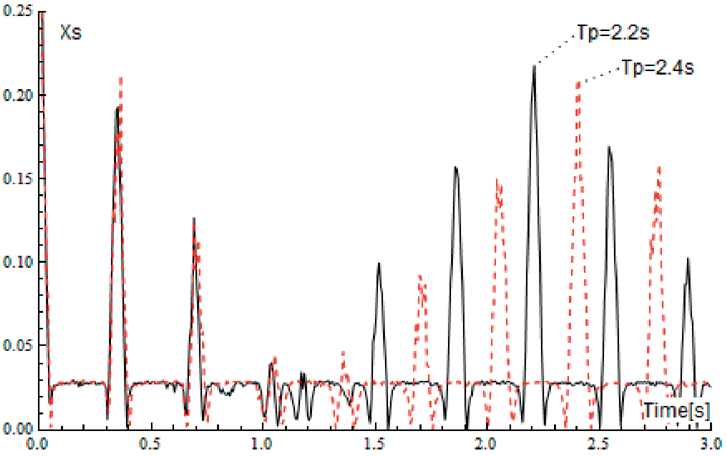


Figure 11. Period detection by active probing

The peaks in correlation values for the two T_p cases (2.2s and 2.4s) are quite clear. The side lobes surrounding the center peak are in line with the correlation results for the theoretical signal as shown in Fig. 6 and 7. The side lobes indicate that the main burst period contains sub-burst periods within it. These sub-bursts could be associated with the underlying TCP mechanisms carrying the HTTP encapsulated video stream. When we focus in on the lower range of the time axis (cf. Fig. 12) we see the same type of side lobes with decreasing peak value.

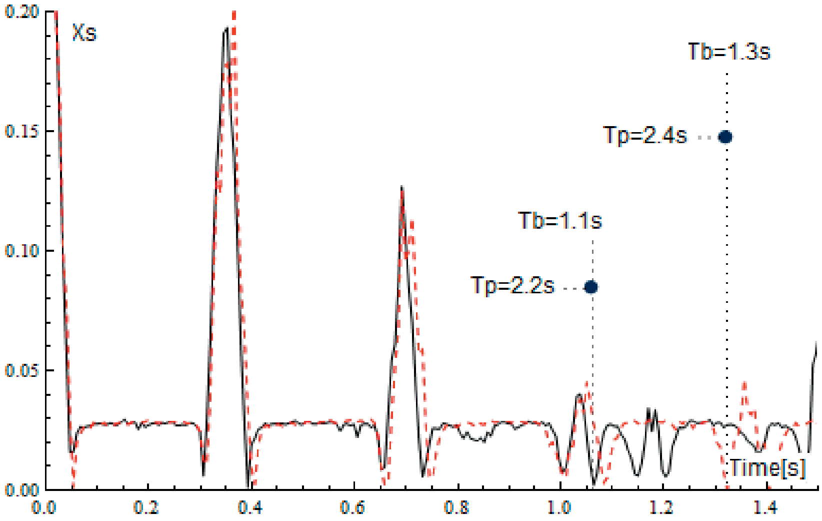


Figure 12. Estimated burst duration by active probing

Following the same approach as for the 100Mbps access case, i.e. searching for the point in time when X_s reaches zero is not enough in this case. The analysis must take all the lobes into consideration, and look for X_s reaching zero after the last side lobe. Obviously, this works well based on the graphical view of X_s but could introduce some challenges in a pure computational approach. However, the potential level of precision in burst duration estimation is quite good.

6.3 Netflix Streaming on 25Mbps and 100Mbps Access

Measurements using the Netflix live stream in both 2Mbps and 3Mbps quality levels were done, over the full range of access capacity levels. The findings were similar for all access capacity levels, except for at the highest level of 100Mbps. In the presentation of the results (cf. Fig. 13) we only show the output from the serial correlation for the 3Mbsp stream, when using access capacities of 25Mbps and 100Mbps.

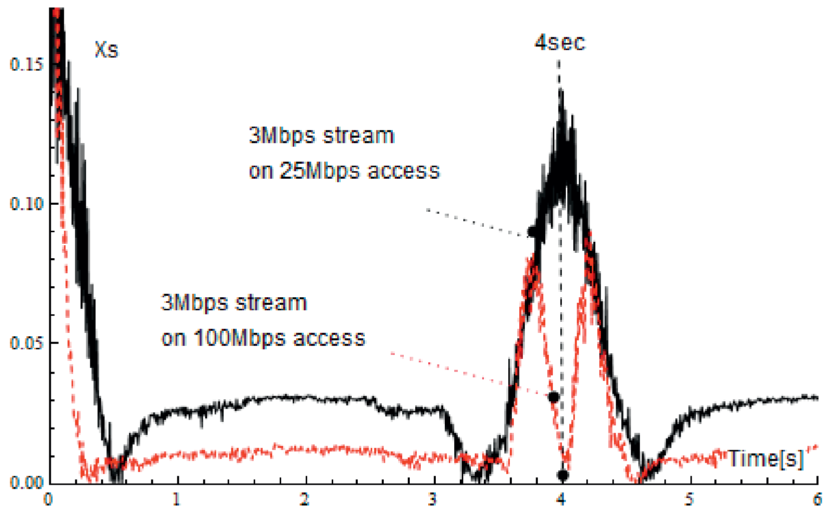


Figure 13. Period detection by active probing

The difference in X_s output noticed at 100Mbps access, is that we instead of a single peak around 4sec get two peaks surrounding this value. Although not studied in depth, the source of this dual peak output is most likely the presence of two TCP sessions. At the highest access level, it seems as if the combined traffic pattern generated by the TCP sessions changes nature. This could be further studied by means of e.g. estimated probability density functions, but this is considered outside of the scope for this paper.

In order to see if we are able to detect the burst duration for a Netflix stream using two TCP sessions, we focus on the lower range of the time axis for the X_s output (cf. Fig. 14). The average values for the real burst durations at 25Mbps and 100Mbps (Ref. Section III) are indicated at 0.45s and 0.21s.

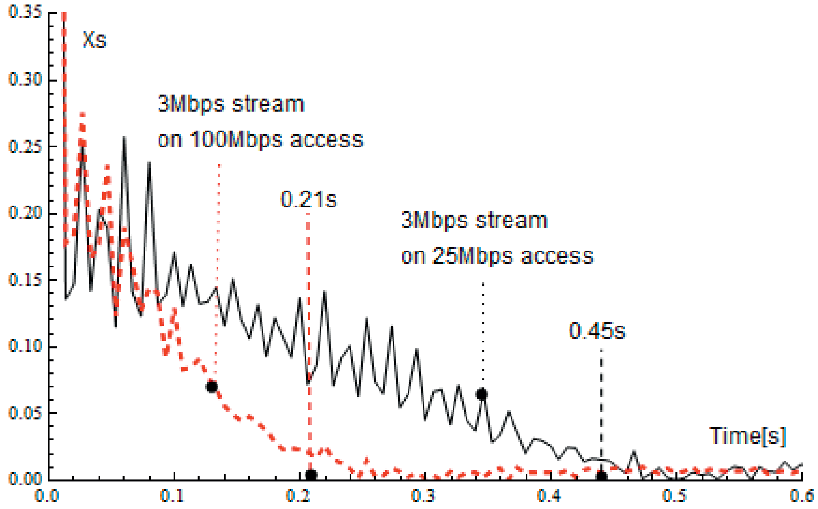


Figure 14. Estimated burst duration by active probing

The X_s output for both 25Mbps and 100Mbps access indicate burst durations of respectively 0.48s and 0.25s. These estimates are a little bit high, but keeping in mind that we are probing the aggregate of two TCP sessions, the results are quite promising.

6.4 Active Probing Rate

The amount of probe traffic required in order to detect and estimate the period and burst durations in an accurate manner is of interest and concern. The default probe pattern used in our experiments (cf. Fig. 5) was a sequence of 100 byte packet pairs, with a fixed 0.5ms gap between the packets (t_{in}) and 6.1ms gap between the packet pairs (t_{gap}). This gives a probe packet rate of 300pps, corresponding to about 240Kbps. However, it is important to note that it takes 2 probe packets (cf. Fig. 5) to produce one sample input to the computation of X_s. Thus, the probing rate is half of the probe packet rate.

To study the effect of changing the amount of probe traffic, we performed experiments with probe packet rates down to 160pps and up to 700pps (cf. Fig. 15), by changing the gap between packet pairs (t_{gap}). The findings for both the video service in lab and Netflix were similar, thus we only present the results for Netflix on a 25Mbps access link as illustration.

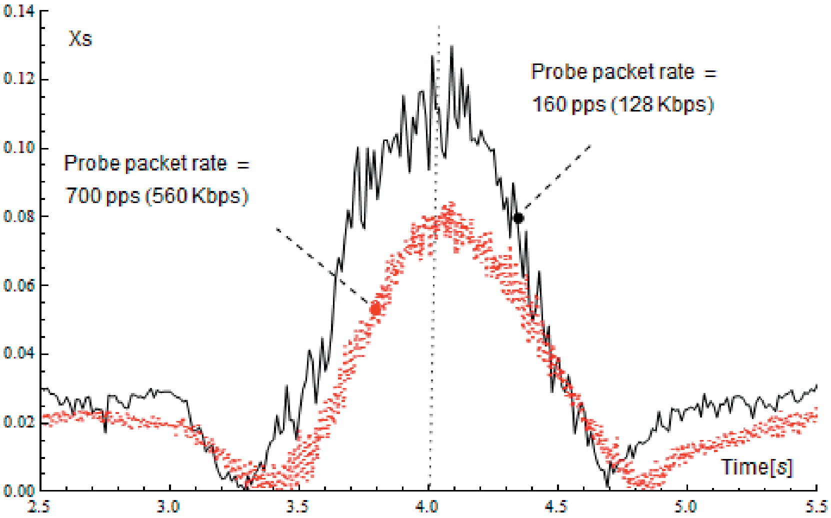


Figure 15. Period detection, High/Low probing

By investigating the serial correlation output Xs for Netflix (3Mbps level) for the purpose of detecting the period, both the lowest and highest probe packet rates give the same result. The period is detected to be around 4sec, as indicated by the peak in the X_s plots. The main difference between the plots used to detect the period is that a higher probe packet rate gives a smoother curve but at the same a less distinct peak.

Focusing in on the lower range of the time axis for the X_s output for respectively high and low probe packet rates (cf. Fig. 16), we see that the same conclusion can be drawn with regards to burst duration based on high and low probe packet rates. However, the higher probe packet rate seems to indicate a higher value for burst duration than the lower. The average value for burst duration, captured using passive measurement (cf. Section III) for Netflix operating at 3Mbps quality level in our measurements was 0.45sec per TCP session. Thus, both the higher and lower probe packet rates give an estimate for the burst duration which is somewhat high, i.e. between 0.55-0.65s.

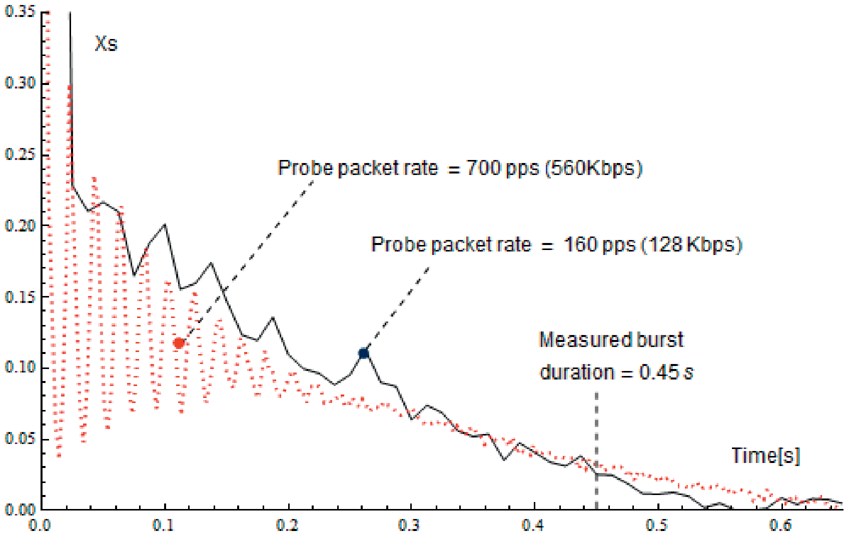


Figure 16. Estimated burst duration, High/Low probing

Keeping in mind that the Netflix stream contains two TCP sessions, and that the active probing is influenced by both of these the results for burst duration detection are considered quite good.

6.5 Sample size

When performing probing of real traffic it is important to reach a state where estimators for the parameters of interest can be presented as fast as possible. In our case, where serial correlation on a time series of observed packet pair gaps is performed, we are concerned about how long the time series must be in order for our estimators of both period and burst duration to appear. Obviously, detecting a periodic pattern in a time series requires us to at least study a time series of length greater than the repeating pattern.

A range of experiments were performed for the video streams available (Smooth Streaming and Netflix) over the full range of access capacities. As illustration of the findings, the results for 5Mbps Smooth Streaming on a 25Mbps access is shown in Fig. 17.

The findings for this specific case was that a sample size of about 10sec was required in order to get a clear peak in the X_s output at the correct time value (T_p=2.0). If the sample size was reduced to only 5 seconds, the peak in X_s output is shifted down to 1.65 seconds, which is not correct.

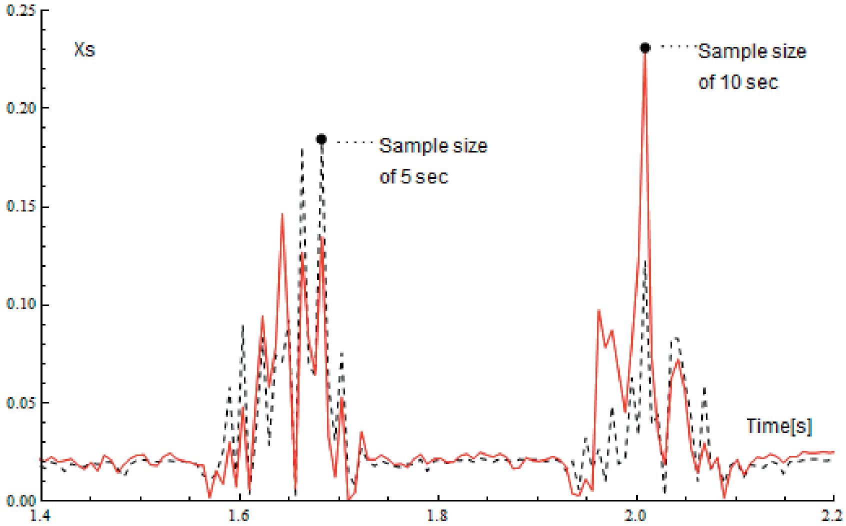


Figure 17. Period detection for different sample sizes

The minimum sample size varies depending on the type of video stream and access capacity, but in all scenarios investigated in our work a sample size of 10 seconds of probe traffic was sufficient for detecting both period and burst duration.

7. Conclusions

The results from our experimental evaluation of the suggested method for detection of periodic behavior and burst duration in cross traffic on access links are considered quite promising. The use of video as the cross-traffic service component in our experiments strengthens the significance of our findings due to the popularity of such services on the Internet.

The use of serial correlation as a tool for analyzing a time series of observations is clearly a strong approach for the purpose of detecting periodic behavior. In all our experiments, the match between results from passive measurements (TCPdump) and active measurements (probing) have been very good. This applies to both the Smooth Streaming lab service, and also the live Netflix service. In the lab scenario, the addition of CBR background traffic does not change these findings.

Analyzing the output of the serial correlation in order to estimate the burst duration is more challenging when the access capacity is reduced, and the side lobes of the serial correlation appears. In these cases, a certain error margin should be expected. However, it is clear that the information is available in the serial correlation output.

The amount of probe traffic required for the method to perform well is considered to be acceptable. The experiments showed that using probe rates down to 128Kbps gave both period and burst information with reasonable accuracy. The results also showed that probing too much did not improve the accuracy to such an extent that it should be considered worth the additional bandwidth.

The required sample size for the method to give estimators for period and burst duration was only subject for a basic evaluation. However, even in this domain the initial findings were positive. A required sample size of around 10 seconds does not directly qualify the method for all purposes. However, one should keep in mind that the periodic components of the services used were in the order of seconds. Therefore, it remains an open issue how fast this type of method can be expected to operate.

8. Future Work

The use of a more complex cross-traffic mix than just video services and CBR traffic is interesting to study. How well our method performs in such cases is an open issue and should therefore be investigated further.

Finding the optimal probe traffic pattern (packet pairs, packet trains etc.) was not subject for study in our work. There is most likely room for further improvements if this topic is investigated in more depth.

In the analytical part of our work we have relied a lot on the manual interpretation of graphical views of the output from serial correlation. If the method is considered for real implementation, some more work must be done in order to develop a more computational oriented approach.

Using this type of cross-traffic characterization as our method provides is also considered interesting to use as input to algorithms for available bandwidth estimations. We plan to investigate this topic for the purpose of developing a new method of performing available bandwidth estimation by means of stratified probing

References

[1] Cisco Systems, “The Zettabyte Era”, Whitepaper, pp. 1-17, May 2012.

[2] I. Sodagar, “The MPEG-DASH standard for multimedia streaming over the internet,” Multimedia, IEEE, vol. 18, no. 4, pp. 62 –67, April 2011.

[3] P. Brockwell and R. Davis, Time Series: Theory and Methods. Springer, vol.1, pp. 1-567, 2009.

[4] P. Lewis and E. Orav, Simulation Methodology for Statisticians, Operations Analysts, and Engineers, vol.1, pp. 1-398, 1988.

[5] R. Yarlagadda, Analog and Digital Signals and Systems. Springer, vol.1, pp. 1-569, 2010.

[6] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, “Bandwidth estimation: metrics, measurement techniques, and tools,” Network, IEEE, vol. 17, no. 6, pp. 27 – 35, nov.-dec. 2003.

[7] E. Goldoni and M. Schivi, “End-to-end available bandwidth estimation tools, an experimental comparison,” in Traffic Monitoring and Analysis, Springer Berlin Heidelberg, 2010, vol. 6003, pp. 171–182.

[8] Y. Cheng, V. Ravindran, and A. Leon-Garcia, “Internet traffic characterization using packet-pair probing,” in INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, may 2007, pp. 1766 –1774.

[9] M. Jain and C. Dovrolis, “Ten fallacies and pitfalls on end-to-end available bandwidth estimation,” in Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, ser. IMC ’04. New York, NY, USA: ACM, 2004, pp. 272–277.

[10] E. Bergfeldt, S. Ekelin, and J. M. Karlsson, “Real-time available-bandwidth estimation using filtering and change detection,” Computer Networks, vol. 53, no. 15, pp. 2617 – 2645, 2009.

[11] G. Maier, A. Feldmann, V. Paxson, and M. Allman, “On dominant characteristics of residential broadband internet traffic,” in Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, ser. IMC ’09. New York, NY, USA: ACM, 2009, pp. 90–102.

[12] H. Jiang and C. Dovrolis, “Why is the internet traffic bursty in short time scales,” in In Sigmetrics. ACM Press, 2005, vol. 1, pp. 241–252.

[13] A. Rao, A. Legout, Y.-s. Lim, D. Towsley, C. Barakat, and W. Dabbous, “Network characteristics of video streaming traffic,” in Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies, ser. CoNEXT ’11. New York, NY, USA: ACM, 2011, pp. 25:1–25:12.

[14] B. J. Villa and P. E. Heegaard, “Group based traffic shaping for adaptive http video streaming,” in Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013), March 2013.

[15] D. R. Figueiredo, B. Liu, V. Misra, and D. Towsley, “On the autocorrelation structure of tcp traffic,” Computer Networks, vol. 40, no. 3, pp. 339 – 361, 2002.

[16] K. Suh, D. R. Figueiredo, J. Kurose, and D. Towsley, “Characterizing and detecting skype-relayed traffic,” in In Proceedings of IEEE INFOCOM ’06, pp. 1-12, April 2006.

[17] B. Kulapala, P. Seeling, and M. Reisslein, "Comparison of traffic and quality characteristics of rate-controlled wavelet and dct video," in ICCCN, 2004, pp. 247–252.

[18] A. Rizk, Z. Bozakov, and M. Fidler, "H-probe: Estimating traffic correlations from sampling and active network probing" CoRR, vol. abs/1208.2870, pp. 1-18, 2012.

[19] B. Villa and P. Heegaard, "Improving perceived fairness and QoE for adaptive video streams," in Proceedings of ICNS 2012, pp. 149-158, March 2012.

[20] B. J. Villa, P. E. Heegaard, and A. Instefjord, "Improving fairness for adaptive http video streaming," in Proceedings of the 18th EUNICE/IFIP, vol.1, pp. 183–193, June 2012,

[21] A. Zambelli, "IIS Smooth streaming technical overview," Tech. Rep., March 2009

[22] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," in ACM Multimedia Systems (MMSys), vol.1, pp. 157-168, Feb. 2011.

[23] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," ACM Trans. Comput. Syst., vol. 18, no. 3, pp. 263–297, Aug. 2000.

[24] J. Laine, S. Saaristo, and R. Prior, "Real-time udp data emitter (rude) and collector for rude (crude)," http://rude.sourceforge.net/, Jan 2000.

PAPER 9: A Measurement Study of Active Probing on Access Links

Bjørn J. Villa, Poul E. Heegaard

Published in

Proceedings of the 19th EUNICE Workshop on Advances in Communication Networking 2013 (EUNICE 2013)

Springer, ISBN 978-3-642-40551-8. pp. 124-135

Aug. 28-30, 2013, Chemnitz, Germany

A Measurement Study of Active Probing on Access Links

Bjørn J. Villa, Poul E. Heegaard

Abstract: This paper presents a measurement study of two different methods for active probing of cross traffic on access links. The categories used in the study are packet pair probing and one-way-delay probing. The first approach uses measured increase in packet spacing as indicator of cross traffic presence, while the latter uses increase in one-way-delay for probe packets as indicator. These methods have been chosen because they are fundamentally different in terms of requirements, benefits and challenges. The main novelty of this paper is the presentation and discussion of measurement results from an access network using an adaptive video service as cross-traffic. The findings clearly illustrate the potential strengths of a probing method based on one-way-delay measurements, under the condition that the required timing accuracy is achieved and delay characteristics is available for the involved network path. The benefit of using Precision Time Protocol instead of Network Time Protocol is illustrated, even in networks of limited size.

Keywords: Packet Pair probing, One-Way-Delay probing, Access links.

1. Introduction

The dynamics of Internet traffic on different levels, ranging from per session, per user and up to backbone traffic aggregates is a topic of great interest in the research community. The reasons for performing such studies are diverse and so are also the techniques and methods applied. One obvious reason for studying Internet traffic dynamics on an aggregated level is the need for knowledge about how to best design and scale the future Internet as a whole. Another reason, on a lower level, is the growing amount of adaptive services on the Internet, which is characterized by their ability to change their requirements according to varying network conditions. Such services would obviously benefit from being able to obtain accurate views of different network- and traffic metrics in real-time.

Obtaining information about Internet traffic is best done by means of passive measurements performed in a non-intrusive way on the network links of interests. However, in some cases such passive measurements are not possible due to e.g. lack of access to the relevant links or involved equipment. In these cases, the use of active measurements is an alternative to consider. Such measurements are based on injecting probe traffic into the network between end-points and then study how this specific traffic is treated. Based on the findings for the probe traffic, one can then make some statements about the traffic conditions along the same path as the probe traffic followed.

Examples of such statements could be related to e.g. packet loss or delay, described by basic mean value considerations or higher order statistical views.

In this paper we compare two active probing methods for estimating cross-traffic amount on access links by means of theoretical discussions and measurements in a controlled lab environment. The metric of interest describing the cross traffic amount is based on buffering time observations for injected probe traffic. These observations can then be analyzed over certain periods and used for different purposes. The methods used in our study are based on sequences of packet pairs and single packets, at different rates. The way buffering time observations are extracted for these two methods are quite different and will be further described later in this paper.

The measurement part of our study is done in a controlled lab environment reflecting a typical broadband access network, and using a high quality video streaming service as the cross traffic component. The reason for focusing on the access link part is that this is where we quite often encounter the bottleneck across a network path. The choice of video as cross traffic component is based on the growing popularity of this service type on the Internet.

The structure of this paper is as follows. Section 2 provides an overview of related work; Section 3 describes the active probing methods; Section 4 presents the measurement setup; Section 5 provides the results and an analysis; Section 6 provides the conclusions and an outline of future work is given in Section 7.

2. Related Work

There is a lot of research in the field of active probing addressing different research questions ranging from the application layer down to the physical layer. In the context of our work, i.e. estimating the amount of cross traffic (measured by increased buffering time for probe traffic) on an access link – the most related research are found in the domain of methods for available bandwidth estimations. For this purpose there are several approaches, most of which fall into either the Probe Rate Model (PRM) or Probe Gap Model (PGM) categories [1]. The PRM approach is based on the principle of self-induced congestion and by this detecting available capacity, while the PGM approach uses observed inter-arrival time (IAT) variations [2][3] for probe packets to estimate the current level of cross traffic. However, the original idea of using probe packets as basis for active measurements was suggested in [4] where back-to-back packets were sent to detect the capacity of bottlenecks.

The use of one-way-delay (OWD) observations [5][6] for probe packets through a network can also be used for estimating available bandwidth as per the PGM approach. However, as the computation of OWD is based on time information from different nodes in the network it has very strict requirements in terms of clock accuracy and synchronization [7]. As described in [8] the main protocols for distributing clock information across as network, NTP (Network Time Protocol) and PTP (Precision Time Protocol) have different capabilities in this regard. The latter is stated to give accuracy in the order of µs, while the former in the order of ms. However, it should be noted that this depends a lot on the specific hardware and software used. In a recent work [9] the performance of the Linux PTP daemon was evaluated and their findings were in line with [8]. What concerns NTP there are also improvements in this provided by the NTPv4 [10] which could bring the accuracy down into the µs region in certain cases.

As presented in [11] there are many sources of delay components along a network path and not all of them are influenced by cross traffic. This represents as source of error for all delay based probing methods.

3. Active Probing

The metric of interest to be measured by the active probing is amount of cross traffic present, represented by introduced additional buffering time for the probe traffic over some time interval. In Fig. 1 a simplified model for an access link as a basic queuing system is presented. The service rate λ_{out} corresponds to access capacity (bits/s), the λ_{cross} corresponds to the uplink capacity for the access node and V_B is the configurable buffer size (bytes) for a specific access. The indicated time parameters t_a and t_b represents time between packets in a packet pair and time between packet pairs as sent, while $t_{a*,i}$ and $t_{b*,i}$ are the corresponding values when probe traffic is received on the client side. The $t_{s,i}$ and $t_{r,i}$ parameters are timestamps for when a packet was sent and received.

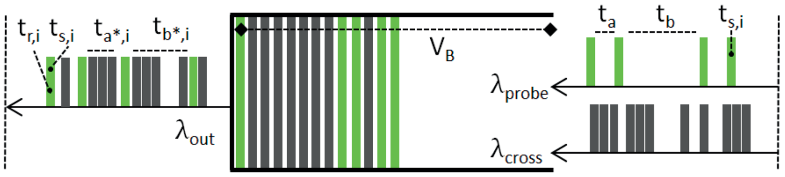


Fig. 1 Access Link Buffer Model

In the zero cross-traffic case the client side will in theory receive the probe traffic with $t_{a*,i} = t_a$, $t_{b*,i} = t_b$ and a constant $t_{s,i}$ - $t_{r,i}$. When cross-traffic is introduced there will be time variations in the received probe traffic, caused by additional buffering time for the probe packets in the access link buffer.

When using packet pairs, the original time between the packets (t_a , t_b) is assumed known by the receiver. Thus, any changes to this would be caused either by cross-traffic or fluctuations in processing load on involved network components. By calculating the difference between packet spacing as received and sent – a series of samples is produced. An important benefit of this method is that there is no need for accurate time synchronization of sender and receiver side. Another benefit is that an increasing amount of probe traffic does not lead to potentially an over-sampling scenario, i.e. the registration of a certain buffering time component more than once. However, the method has a weakness in the sense that cross-traffic may delay the first packet in a pair, and thereby reduce the spacing between the packet pairs [12]. One way to handle this is to consider both packet pair spacing and the time between packet pairs by summarizing this into packet pair period samples $t_{pp,i}$ as given in Eq.1.

$$t_{pp,i} = \left[t_{a^*,i} - t_a\right]^+ + \left[t_{b^*,i} - t_b\right]^+ \qquad (1)$$

In addition, some computation is required on the period sample time series as presented in our submitted work [13] in order to carry forward a time shift component to the next $t_{a*,i}$ or $t_{b*,i}$ observation. However, for the purpose of comparing the two active probing methods in this paper we have left this computational correction out. It is further important to note that for this method to be able to capture delay components higher than just the buffer output time of a single cross-traffic packet T_p, it is a requirement that the arrival rate λ_{cross} towards the bottleneck is higher than the service rate λ_{out}. This can be seen from Eq. 2 where the maximum value for observed time between packets in a packet pair $t_{a*,Max}$ is expressed.

$$t_{a^*,Max} = \frac{(\lambda_{cross} t_a)}{\lambda_{out}} + T_p, \quad for \quad t_a \leq V_B/\lambda_{out} \tag{2}$$

In a real life network the condition $\lambda_{cross} > \lambda_{out}$ would normally apply since an access node typically is served by at least a gigabit connection and each customer connection would be in the order of tens of Mbps. One could also claim that the approach of using packet pairs has a drawback in the sense that is requires two probe packets to produce a single cross traffic sample. However, if both packet pair spacing and time between packet pairs are considered, this is no longer applicable.

When using a sequence of single probe packets it is the OWD for each packet which is used to obtain buffering time samples. The sender adds a time stamp to each packet when sent $t_{s,i}$, and the receiver adds his own timestamp to the packet when received $t_{r,i}$. If we then know the reference OWD during times of zero cross traffic $T_{owd,0}$, we can for each probe packet when received - compute a sample $t_{owd,i}$ for buffering time induced by cross-traffic.

$$t_{owd,i} = \left[(t_{r,i} - t_{s,i}) - T_{owd,0}\right]^+ \tag{3}$$

This approach has the benefit of that each probe packet gives one cross-traffic sample, and all samples are independent. However, even though the samples are independent they may actually lead to a degree of oversampling if more than one probe packet is in the buffer at the same time. The reason is that each probe packet will be delayed according to the total amount of packets ahead of it in the buffer, even if there are other probe packets as well there. Investigation of the over-sampling issue is left for future work. Further on, as the method uses timestamps from different sources (sender and receiver) it requires a high degree of accuracy in time synchronization. The use of NTP or even PTP may not be accurate enough. The challenge of actually knowing the reference OWD when no cross-traffic is present is also a significant challenge.

4. Measurement Setup

In order to perform a comparison of the two active probing methods, an access network testbed was established (cf. Fig. 2). In order to minimize cross process impacts on the client and server side, both the probe generator and the probe receiver were put on dedicated nodes. In a real life, this may be more integrated at least on the client side – but this depends on the specific application.

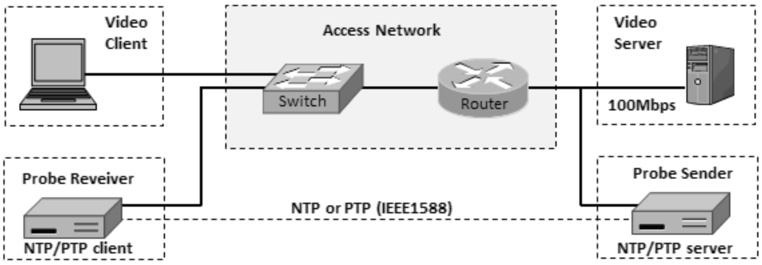


Fig. 2 Access Network Testbed

As probe traffic generator and receiver the Rude/Crude tool [14] was used. This has the capability of generating IP packets according to trace files, describing both packet size and time between packets. It also provides application level time stamping which is easily available on the receiver side. The accuracy of this tool has been shown in [15] to be in the area of 2µs. The cross-traffic used in the measurements was a video stream operating at 5Mbps based on the MS Smooth Streaming platform [16]. In our earlier work [17] the nature of this traffic is described in more detail with special focus on its burst oriented nature. Based on this, we can state that a video server of this type connected on a 100Mbps link will send bursts of data towards the client at rates close to its link speed, independent of what the average video stream bitrate is. Thus, we have that the earlier stated $\lambda_{cross} > \lambda_{out}$ requirement for packet probing method is met. The access capacity towards the client was configured using the QoS mechanisms provided by the Cisco switch used. This configuration gives the λ_{out} and also the buffer size V_B available for the specific access.

For the purpose of time synchronization of probe sender and receiver both NTPv4 and PTPv2 [18] were used in the measurements. The NTP configuration was made so that the probe receiver used the probe sender as NTP server, in order to maximize timing accuracy when using this protocol. When using PTP as synchronization protocol, the same direct relationship between the probe sender and receiver was made. The probe sender in PTP master role and the probe receiver in PTP slave role.

The operating systems used on the video client and server side were MS Windows 7 Professional, while probe sender and receiver were using Linux Ubuntu 12.04.

All processing of measurement data was done post-experiment in order to keep the cross-process impact for each node as low as possible.

4.1 Measurement scenarios

A range of different probe traffic patterns was used in the measurements. The parameters subject for change were the intra-packet time values t_a and t_b, while the probe packet size was fixed at 100Byte in all cases. When using packet pair probing, t_a was always smaller than t_b, thus reflecting the time between packets in a pair. In the case with a sequence of single probe packets and measurement of OWD, t_a was set equal to t_b. The sum of t_a and t_b gives the period in the probe pattern and thereby also the probing rate in pps and bps. Ideally, the probing rate should be kept as low as possible, in order to minimize the chances of self-induced congestion or other undesirable service impact.

Table 1. Configuration parameters

V_B [MB]	Packet Pair			One-Way-Delay		
	t_a [ms]	t_b [ms]	pps	t_a [ms]	t_b [ms]	pps
256/500/1000	0.55/0.65/0.75	4.6/4.5/4.4	388	2,60	2,60	383
256/500/1000	0.55/0.65/0.75	3.7/3.6/3.5	470	2,13	2,13	468
256/500/1000	0.55/0.65/0.75	3.0/2.9/2.8	562	1,78	1,78	559
256/500/1000	0.55/0.65/0.75	2.5/2.4/2.3	654	1,53	1,53	650
256/500/1000	0.55/0.65/0.75	2.1/2.0/1.9	752	1,33	1,33	747

The parameters given in Table 1 represent the range of different scenarios included in our measurements, for which we compared the two methods of active probing. For each scenario, measurements were done for a period of 10 minutes both with and without the cross-traffic (i.e. the 5Mbps video stream). The capacity on the access link was set to 10Mbps for all scenarios, but with different buffer (V_B) settings configured in the router.

5. Results

In this section, a selection of the measurement results is presented. The specific scenario for packet pair probing where V_B=256/t_a=0.55/t_b=4.6, and for OWD probing where V_B=256 / t_a=t_b=2.6 is presented in detail. The presentation of the results are mainly given by means of graphical summaries, and especially by estimated probability density function (PDF) plots and the corresponding cumulative distribution function (CDF). The differences in distributional properties for the received probe traffic with and without cross-traffic present are quite well presented by this. Whenever appropriate, interesting numerical indicators are also included.

The effect of time synchronization method used (NTP, PTP) is only presented for the OWD probing method. The reason for this is that the packet pair method only uses receiver side time information, and therefore is not affected by this.

5.1 Packet Pair probing results

For the packet pair probing method, the difference between the PDF for probe packets received when there is no cross-traffic present, and when the video stream is introduced is clear in terms of the reduced distribution peak for the latter case. In addition, it is interesting to note the appearance of a small peak in the distribution for $t_{a*,i}$ (cf. Fig. 3, left side) in the low value region in the case when cross-traffic is present. The source of this effect is the occurrence of $t_{b*,i} > t_b$ samples, which delays the first packet in the next packet pair – as discussed in section 4. The result of this is the low value group of $t_{a*,i} < t_a$ observations.

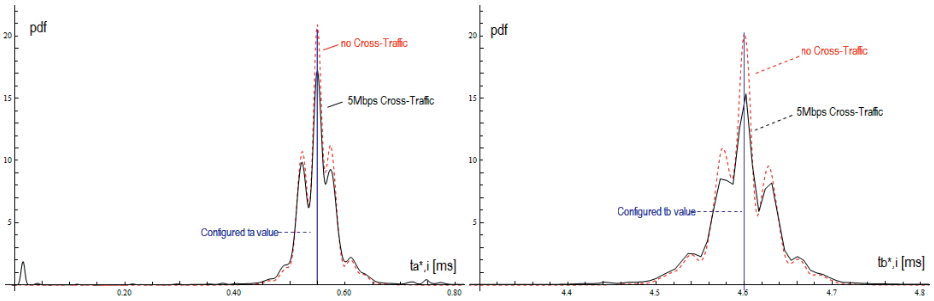


Fig. 3. PDF view of Packet Pair Results at ta=0.55ms, tb=4.6ms and VB=256

By viewing the same measurements using a CDF plot rather than PDF, it is even easier to see that the packet pair probing is able to detect the cross-traffic (cf. Fig. 4). The CDF for $t_{a*,i}$ observations are lifted in the low region, and reduced in the high region. The similar effect is also seen in the CDF for $t_{b*,i}$ observations. Thus, both $t_{a*,i}$ and $t_{b*,i}$ observations detect cross-traffic, but at the same time they have a negative impact on each other (as indicated by the low region CDF lift).

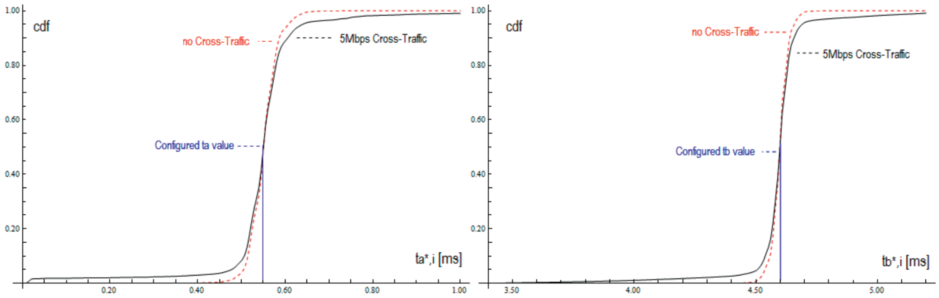


Fig. 4. CDF view of Packet Pair Results at ta=0.55ms, tb=4.6ms and VB=256

Common for both the PDF and CDF view of the packet pair method is that even in the no Cross-Traffic scenario the received probe traffic has some deviations from the original pattern. This represents a significant source of error, which could be critical depending on how the results of the probing are to be applied.

5.2 One-Way-Delay probing results

For the OWD probing method, the difference between the PDF for probe packets received with and without cross-traffic present is quite significant in terms of shape (cf. Fig 5, left side). The dominating peak is shifted upwards when cross-traffic is present, which also contributes to a higher mean value.

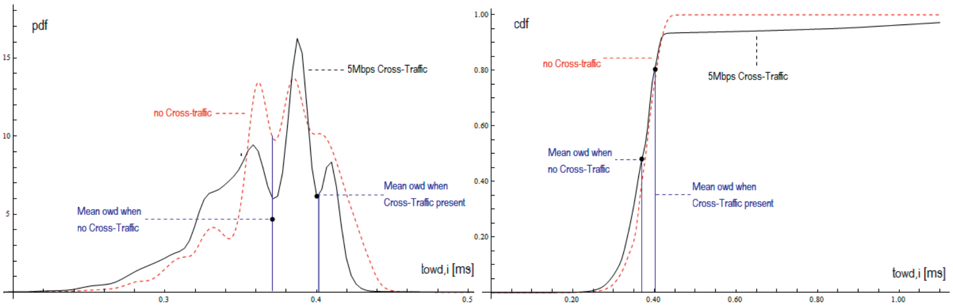


Fig. 5. PDF and CDF view of OWD results at ta=tb=2.6ms, VB=256 and NTP

The effect on the CDF (cf. Fig. 5, right side) illustrates the difference even better, as the two graphs follow each other up to the ~90% level and then the cross-traffic graph flattens out. This indicates that the high ~10% amount of values differ significantly in magnitude. Same as for the packet pair probing method, both the PDF and CDF view of the measurements show variations even in the no Cross-Traffic scenario.

5.3 Effect of NTP versus PTP on One-Way-Delay results

The effect of using PTP as time synchronization protocol between the probe sender and receiver instead of NTP can be illustrated by a PDF and CDF plot for OWD observations as given in Fig. 6.

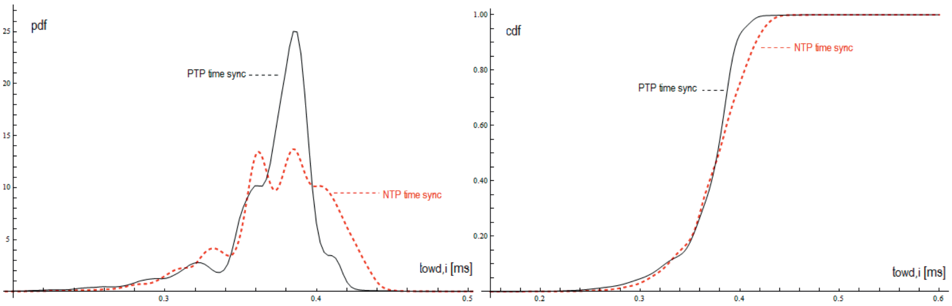


Fig. 6. PDF and CDF view of OWD results at ta=tb=2.6ms and no Cross-Traffic, NTP and PTP

The use of using PTP instead of NTP clearly gives a sharper peak in the PDF function and an increased function derivate in the mid region of the CDF function. These are both indications of more accurate sync. The difference in mean value for the observations when using NTP and PTP is about 6µS, which may increase the error in each probe sample value collected.

5.4 Cross traffic amount detection

The active probing methods generate samples for cross-traffic amount by measuring additional buffering time for the probe traffic caused by the cross-traffic. These samples are generated by either looking at the probe packet pair IAT or probe packet OWD. The difference between measured IAT or OWD values and their corresponding reference values generate a time series of buffering time samples ($t_{pp,i}$ or $t_{owd,i}$). By summarizing the time series for each method, over a 10 minute measurement period while the video stream is at 5Mbps, we get a view on each methods capability of cross-traffic amount detection (cf. Fig. 7).

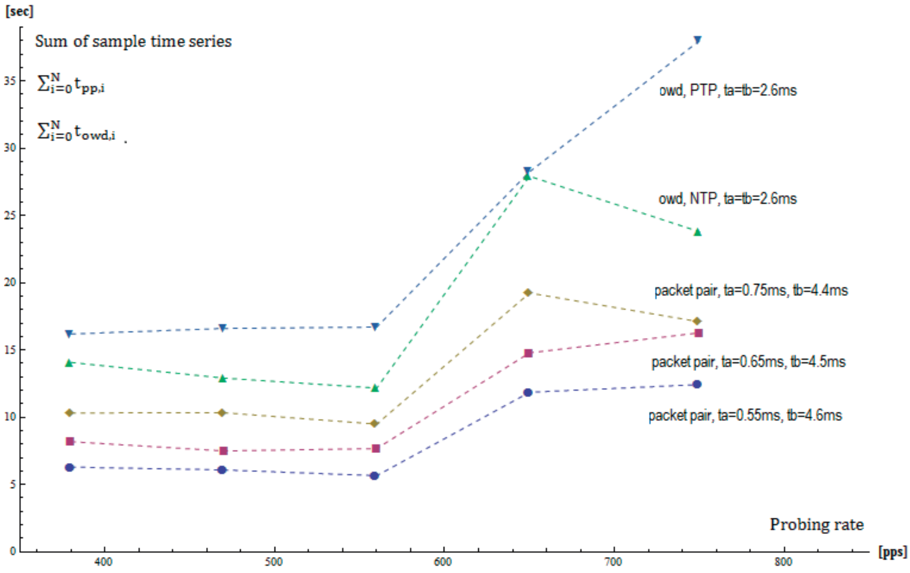


Fig. 7. Comparison of probing methods

We can see that the method based on OWD measurements detects a higher cross-traffic amount for all probe rate levels, than the method based on packet pairs. It is also worth noticing that the two highest probe rate levels give distinct higher results for both methods. The amount of cross-traffic detected does not have the same linear increase as for the probing rate, but instead it seems to cross a threshold between the third and fourth probing rate level used. Although not further investigated, this threshold may indicate that the probe traffic has crossed a level where TCP mechanisms in cross-traffic are affected and thereby the nature of the traffic is changing.

For the packet pair method and the different t_a values used (cf. Table 1) there are also quite noticeable differences. The highest t_a value gives the higher cross-traffic amount detection across all probe rate levels.

For the OWD based method it is clear that the use of PTP for time synchronization instead of NTP has a significant effect, as it always gives a higher cross-traffic estimate. The graph for the PTP case also has a more logical profile than in the NTP case, as the cross-traffic volume consistently increases with increasing probing rate. The NTP graph starts with a moderate decrease, followed by a significant peak before dropping again. Repeated experiments shows that the results for the NTP case fluctuate more than in the PTP case, which is expected due to clock drift.

The presence of potential oversampling for the OWD measurements was not investigated. However, as the trend for both packet pair probing and OWD probing is quite similar across all probing rate levels we believe that this error factor did not contribute much to the measurement results.

6. Conclusions

The measurement based comparison of packet pair and OWD probing presented in this paper highlights strengths and weaknesses for both methods. The objective of the comparison was not to make a statement about which one is better, as this would have to be done in the context of a specific application. However, in a scenario such as the one we used in our measurements, it is clear that the OWD based method is able to detect more of the cross-traffic than the packet pair method.

Concerning the packet pair method, we believe that our findings related to dependencies between t_{a*} and t_{b*} observations have significance for a range of suggested probing methods based on the packet pair principle, e.g. in the area of available bandwidth estimation. To our knowledge, this specific dependency has not been documented earlier. We further believe that our findings related to sensitivity for parameter values t_a and t_b selected are of interest. Finding the optimal parameter selection for a complex traffic mix is foreseen to be quite challenging, but an approach where a range of values are used may be a beneficial approach.

The most interesting finding for the OWD method is the significant impact of using PTPv2 instead of NTPv4. Keeping in mind that our access network lab is of a much smaller size than what a real network would be, it clearly demonstrates the shortcomings of using a NTP based synchronization for purposes like this. Another interesting finding is the challenge to establish a reference point for OWD when no cross-traffic is present. The variations observed even in the small access network lab were higher than what we expected.

7. Future Work

To further analyze the capabilities of active probing methods it would be interesting to also make measurements using other tools than Rude/Crude, and also use different operating systems and even HW components. Reason being that one can never neglect the possibilities when doing measurements that some of the things which are being observed have underlying reasons not related to the topic investigated.

The implementation of time synchronization between probe sender and receiver should also be closer investigated. The specific implementation of NTPv4 in Ubuntu 12.04 may have flaws, or there could even be some not obvious configuration options with a positive impact on accuracy. In a scenario with at least µs accuracy of time synchronization between hosts on the Internet, active probing using OWD measurements becomes very attractive.

The cross-traffic used in our measurements was of a specific type, operated at a specific quality (bitrate) level. A more composed and potentially complex cross-traffic profile would also be interesting to include in a measurement study. However, the burst oriented nature of the video service used gives a very challenging traffic pattern. Thus, we do not think a more composed cross-traffic scenario will make the research question significantly harder.

References

[1] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, “Bandwidth estimation: metrics, measurement techniques, and tools,” Network, IEEE, vol. 17, no. 6, pp. 27 – 35, nov.-dec. 2003.

[2] J. Strauss, D. Katabi, and F. Kaashoek, “A measurement study of available bandwidth estimation tools,” in Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, ser. IMC ’03. New York, NY, USA: ACM, 2003, pp. 39–44.

[3] E. Goldoni and M. Schivi, “End-to-end available bandwidth estimation tools, an experimental comparison,” in Traffic Monitoring and Analysis, ser. Lecture Notes in Computer Science, Springer, 2010, vol. 6003, pp. 171–182.

[4] S. Keshav, “A control-theoretic approach to flow control,” in Proceedings of the conference on Communications architecture & protocols, ser. SIGCOMM ’91. USA: ACM, 1991, pp. 3–15.

[5] M. Jain and C. Dovrolis, “Pathload: A measurement tool for end-to-end available bandwidth,” in In Proceedings of Passive and Active Measurements (PAM) Workshop, 2002, pp. 14–25.

[6] Y.C. Huang, C.S. Lu, and H.K. Wu, “Available bandwidth estimation via one-way delay jitter and queuing delay propagation model,” in Wireless Communications and Networking Conference, 2006. Vol. 1, 2006, pp. 112–121.

[7] M. Shin, M. Park, D. Oh, B. Kim, and J. Lee, “Clock synchronization for one-way delay measurement: A survey,” in Advanced Communication and Networking, Communications in Computer and Information Science, Springer, 2011, vol. 199, pp. 1–10.

[8] L. De Vito, S. Rapuano, and L. Tomaciello, “One-way delay measurement: State of the art,” Instrumentation and Measurement, IEEE Transactions, vol.57, no.12, pp. 2742–2750, 2008.

[9] T. Kovacshazy and B. Ferencz, “Performance evaluation of ptpd, a ieee 1588 implementation, on the x86 linux platform for typical application scenarios,” in Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International, 2012, pp. 2548–2552.

[10] D. Mills, J. Martin, J. Burbank, and W. Kasch, “Network Time Protocol Version 4: Protocol and Algorithms Specification,” RFC 5905 (Proposed Standard), IETF, Jun. 2010.

[11] A. Hernandez and E. Magana, “One-way delay measurement and characterization,” in Proceedings of the Third International Conference on Networking and Services, ser. ICNS ’07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 114.

[12] N. Hu, S. Member, P. Steenkiste, and S. Member, “Evaluation and characterization of available bandwidth probing techniques,” IEEE Journal on Selected Areas in Communications, vol. 21, pp. 879–894, 2003.

[13] B. J. Villa and P. E. Heegaard, “Estimating available bandwidth on access links by means of stratified probing,” in Proceedings of the 6th International Conference on Computer and Electrical Engineering Conference (ICCEE 2013), Oct 2013.

[14] J. Laine, S. Saaristo, and R. Prior, “Real-time udp data emitter (rude) and collector for rude (crude),” http://rude.sourceforge.net/, Jan 2000.

[15] S. Ubik, V. Smotlach, S. Saaristo, J. Laine, “Low-cost precise qos measurement tool”, CESNET Tech report number 7/2001.

[16] A. Zambelli, “IIS smooth streaming technical overview,” http://www.microsoft.com/silverlight/, Tech. Rep., March 2009.

[17] B. J. Villa and P. E. Heegaard, “Group based traffic shaping for adaptive http video streaming,” in Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013), March 2013.

[18] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems 1588-2008, IEEE Std. 1588, 2008.

PAPER 10: Estimating Available Bandwidth on Access Links by Means of Stratified Probing

Bjørn J. Villa, Poul E. Heegaard

Presented at
The 6th International Conference on Computer and Electrical Engineering
(ICCEE 2013)
October. 12-13, 2013, Paris , France

Published in
International Journal of Electrical Energy
ISSN 2301-3656, volume 1 (4), 2013, pp. 213-221

Estimating Available Bandwidth on Access Links by Means of Stratified Probing

Bjørn J. Villa, Poul E. Heegaard

Abstract: This paper presents a novel approach for estimation of available bandwidth on access links using stratified probing. The main challenges of performing such estimations in this network part is related to the bursty nature of cross-traffic and the related uncertainty regarding appropriate time period for producing sample estimates. Under the fluid flow traffic model assumption, these problems would not be present – but for the access network part this assumption does not hold. The method suggested in this paper is based on a four-phase approach. In the first phase a traffic profile for the cross-traffic is established, with focus on detecting periodic behavior and duration of respectively burst and idle sub-periods. In the second and third phase, the active probing is split into strata and synchronized according to the burst/idle sub periods. In the final phase, the actual probing and estimation of available bandwidth takes place. The method is analyzed by means of experiments in a controlled lab environment, using adaptive video streaming as the service with a periodical behavior. The empirical results are considered quite promising both in terms of accuracy and the low degree of intrusiveness facilitated by the stratified approach.

Keywords: Available Bandwidth Estimation, Stratified Probing, Access Network, Adaptive Video Streaming, Over-The-Top.

1. Introduction

The amount of services provided to Internet users around the world following an Over-The-Top service delivery model is increasing. This model is based on that the involved network operators are not taking any active measures in order to assure the quality levels of the specific services. Traffic is carried as part of the best-effort class and will therefore face obvious challenges in terms of being able to meet the end users expectations regarding Quality of Experience (QoE).

In terms of service usage, there has also been a significant change over the last decade. The dominance of services including video components is increasing. This applies both in terms of traffic volume and service usage frequency. The obvious example of such a service provider is YouTube.

For services with video components, the capacity requirement in terms of bitrate is critical as this could be in the order of several megabits per second. Up until a few years ago, delivering such services Over-The-Top was almost impossible without involving the network operators in order to gain access to other Quality of Service (QoS) classes than just best effort. As solutions for adaptive video streaming emerged and the relevant MPEG-DASH standard [1] was approved, this situation changed. By introducing additional functionality in service endpoints, these solutions make it possible to change quality level during service delivery based on certain observed parameters. Among these parameters are available bandwidth between client and server.

The specific methods for estimating available bandwidth and other interesting metrics are not part of the MPEG-DASH standardization effort. This makes it an area of particular interest for technology vendors, in their effort to make their solutions perform well in the market place. However, accurate methods for estimating available bandwidth are of interest also for other services. An interesting aspect in this regard is that the growing amount of video service on the Internet makes it even harder than before to perform available bandwidth estimations due to the embedded bursty nature in traffic generated by these services. This follows by the repeated fetch-next-video-segment operations, and associated traffic bursts from server to client.

The relevance of the research presented in this paper can be viewed as a contribution to the domain of adaptive services in general, which has the need for an indicator on how much cross-traffic is present, and subsequently available bandwidth. In this context, the term cross-traffic is used to describe the aggregate of traffic present on an access link. A continuous estimate of cross-traffic volume can be used for both adjusting quality levels on a per service basis and as input to a network qualification test prior to service usage.

1.1 Problem Statement

The amount of cross-traffic present on an access link during a specific interval (*i),* ending at time *t* is described in terms of number of bits sent (V_i) across it during the time interval (T_p). The available bandwidth ($B_{i,Tp}$) during the same interval is the difference between the total link capacity (*C*) and the cross-traffic estimate (cf. Eq. 1).

$$B_{i,Tp} = C - \frac{1}{T_p} V_i[t - T_p, t] \qquad (1)$$

The sequence of available bandwidth estimates results in a time series B_{Tp} with index *i* over the index set *N*. The size of *N* corresponds to the duration of the available bandwidth estimation (cf. Eq. 2).

$$B_{Tp} = \{B_{i,Tp}, i \in N\} \qquad (2)$$

Using active probing as the method to obtain estimates for the cross-traffic V_i requires a number of *K* probe packets (or packet pairs) to be sent and analyzed by the receiver during each time interval T_p. The results from the probing, gives a sequence of cross-traffic samples v_j indicating how many bits of cross-traffic a probe was influenced by. The cross-traffic influence is detected by measured increase in inter-arrival time for probe packets, or delayed arrival of single probe packets. In order to get the total cross-traffic estimate in number of cross-traffic bits V_i observed for the specific period i, the v_j samples are summarized (cf. Eq. 3).

$$V_i = \sum_{j=1}^{K} v_j \qquad (3)$$

Even though the formulas for this type of estimation are simple, providing input to them which gives an accurate result is not straightforward in cases where the cross-traffic does not follow a fluid-flow model. As will be described closer in Section II, real traffic does not follow the fluid-flow model, and in particular services with video components generate a very bursty traffic profile.

Given the bursty nature of popular services such as video streaming, and put into the context of access networks, it raises some specific challenges for performing accurate available bandwidth estimations. The problem at hand is how to perform active probing of burst oriented cross-traffic without introducing access link congestion, and also how to choose the appropriate time period T_p for computing $B_{i,Tp}$ samples.

Choosing the appropriate time interval for cross-traffic estimation can be done in different ways. From the perspective of describing the cross-traffic as accurate as possible, the time interval should be small enough to cover real-time fluctuations, but at the same time large enough so that it provides useful input to the user (e.g. an application) of the available bandwidth estimations.

It is also important to be aware of that the configuration of access capacities in commercial networks quite often allow traffic bursts in excess the specified capacity *C*. Thus, in small time scales the actual bitrate on the link level will be higher than *C*. If this in not taken into consideration when choosing T_p the resulting B_{Tp} time series will contain occurrences of negative values for available bandwidth.

Our approach to these challenges is a method based on performing stratified probing of the cross-traffic according to its detected profile. The rationale behind a strata oriented probing approach is to maximize the value of each probe packet sent by probing more during bursty cross-traffic periods than during almost idle periods.

The current version of our method is only applicable if the cross-traffic profile has a periodic component of significance. This type of cross-traffic is quite common due to the growing amount of services with video components on the Internet. The periodic and burst oriented nature of such services is described in more detail in Section II. In the case where cross-traffic does not have a periodic component of significance, and alternative probing approach should be considered.

1.2 Research Approach

In order to study both the feasibility of estimating available bandwidth through the use of stratified probing and also how well it performs, we chose an empirical research approach. To support this we established a hybrid active/passive measurement testbed which will be further described in Section V. The passive measurements are used to capture the real traffic profiles and available bandwidth, while the active measurements reflect the results when using our experimental implementation of the investigated method. The main service component used in our research as cross-traffic is adaptive video streaming which generates a typical periodical and burst oriented traffic profile. More details regarding adaptive video streaming will be provided in Section III.

1.3 Paper Outline

The structure of this paper is as follows. Section II provides an overview of related work; Section III provides characteristics of adaptive video streaming; Section IV describes our method of performing available bandwidth estimation; Section V describes our measurement setup; Section VI presents the results; Section VII presents a brief discussion; Section VII presents our conclusions and an outline of future work is given in Section IX.

2. Related Work

There are several approaches for estimating bandwidth along a network path, most of which fall into either the Probe Rate Model (PRM) or Probe Gap Model (PGM) categories [2]. The PRM approach is based on the principle of self-induced congestion and by this detecting available capacity, while the PGM approach uses observed inter-arrival time variations for probe packets to estimate the current level of cross traffic, which then combined with knowledge about the total capacity, can be used to produce an estimate of the currently available capacity. The stratified probing approach described and analyzed in our work, belong to the PGM category.

Within the range of PGM methods published over the last ten years, there are quite a few different versions [2] [3] [4] [5] [6]. They all use the principle of inserting probe packets in such a way that they follow the same path as the cross traffic of interest. However, when it comes to how many probe packets per time unit and patterns of such there are many differences.

In terms of how well the existing methods perform both in terms of accuracy and real-time capabilities a few studies have been published [7]. Early work in this area showed that a Poisson approach of spacing probe packets gave a significant improvement over the fixed approaches. More recent work [8] has documented the need for careful calibration of methods used in order for them to perform as good as possible. For the special case of available bandwidth estimation on access links, where the cross-traffic contains burst components – we have not been able to find any relevant research published. .

With regards to understanding the nature of adaptive video streaming and resulting traffic, we studied this from different perspectives in our earlier work [9] [10]. Further on, in [11] we presented a new method for achieving a traffic shaping effect for adaptive video streaming, without involvement from network components. The purpose of this was to make the traffic easier to characterize by probing methods, by reducing the degree of burstiness in traffic.

In the first phase of the method we are presenting, we use a technique for detecting period and burst duration we earlier [12] proposed. This method uses serial correlation [13] on time series of observations in order to detect periodic properties in traffic. Such properties are commonly found in TCP traffic in general and in video streaming services in particular.

3. Adaptive video streaming

As stated in the introduction part there are many approaches to adaptive video streaming, and a new standard [1] has emerged in this domain. The specific solution used as basis for our research is the Smooth Streaming Solution from Microsoft [14]. The periodic nature of adaptive video streaming is given by the repeated requests for the next segment in a video stream, at a specific quality level (cf. Fig. 1).

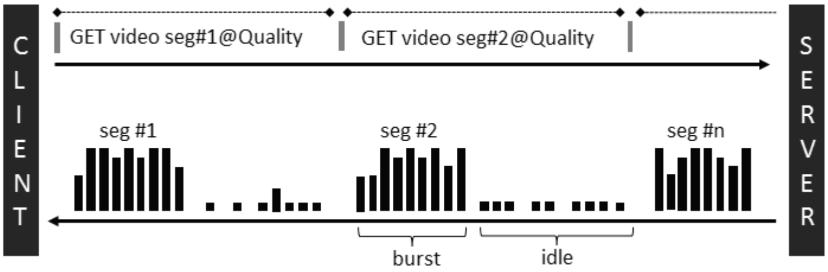


Fig. 1 Traffic profile for adaptive video streaming

The frequency of these requests differs between vendor solutions in general, and it is also to some extent dependent of implementation choices. The Smooth Streaming solution used in our experiments has a default GET segment request interval of 2 sec, which then would represent the period of interest (T_p) to identify for our probing method. The duration of the burst periods (time to get next video segment) varies depending on current quality level of the video stream, the access capacity and also the GET segment request interval. The resulting dynamics in duration of the burst period (T_b) is illustrated in Fig. 2. The presented values are based on passive measurements of a single video stream, operating without any other traffic present on the access link.

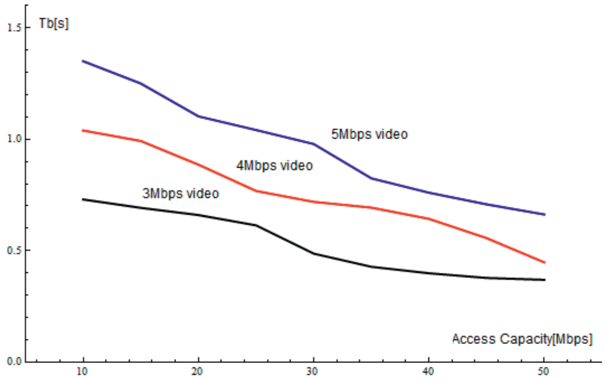


Fig. 2 Passive measured burst period duration

The presence of bursty traffic as described represents a challenge for both active and passive measurements. Passive measurements are of course easier as they do not have the need for any type of probing. However, even for passive measurements the resulting view of traffic load on an access link looks very different depending on over which period the average is calculated. In Fig. 3 we show the average bitrates for an adaptive video streaming service running at 4Mbps quality level, and the default T_p value of 2 sec.

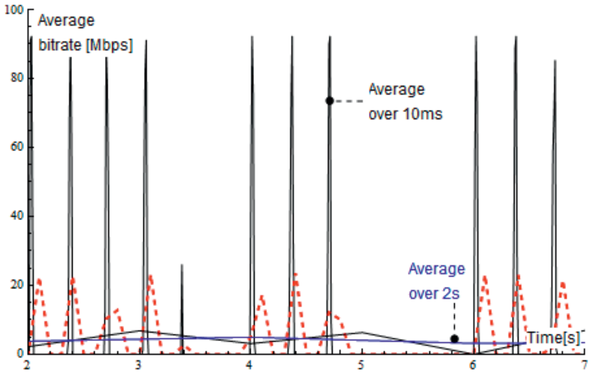


Fig. 3 Passive measured average bitrates over different periods

The different lines in Fig. 3 give the results when using time intervals for average bitrate measurements in the range from 10ms to 2sec. The smallest time interval is able to capture the short lived traffic spikes up to about 90Mbps, while the largest time interval gives a more accurate view on the actual service quality level of about 4Mbps.

Comparing the passive measured bit rates in Fig. 3 with the traffic profile for adaptive video streaming in Fig. 1, the similarities are clear. During the burst periods there are a number of traffic spikes, while during the idle periods these spikes are not present. The size and number of spikes are closely related to streaming server capacity, physical link rate and basic TCP mechanisms. The specific TCP version used will also have an impact on the traffic pattern inside a burst period.

4. Method

The suggested method for estimating available bandwidth is composed by different phases. How an actual implementation of the method will pass through these phases depends on the actual application. In a continuous estimation process of available bandwidth and a dynamic cross-traffic picture, there is a need to re-visit the first phase after some time. In the method flowchart (cf. Fig. 4) this is indicated by the return to first phase after *n* periods.

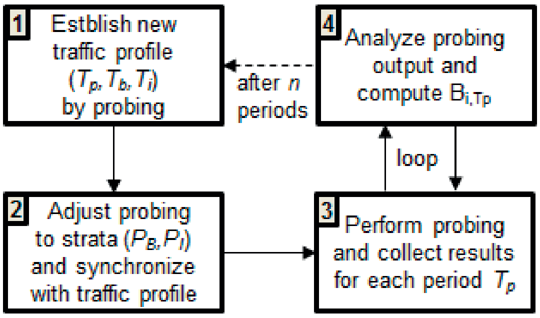

1 Estblish new traffic profile (T_p, T_b, T_i) by probing
2 Adjust probing to strata (P_B, P_I) and synchronize with traffic profile
3 Perform probing and collect results for each period T_p
4 Analyze probing output and compute $B_{i,Tp}$
after *n* periods
loop

Fig. 4 Phased approach for estimating available bandwidth

In the first phase, a traffic profile for the cross-traffic is established by use of active probing. This profile is used as input to the next phase where new probe rates are decided for each of the two strata (burst and idle sub-periods), and a synchronization of the probe strata and cross-traffic sub-periods are done. In the third phase the active probing of cross-traffic is performed in order to obtain v_j samples (cf. Eq. 3). In the last phase the available bandwidth is estimated.

4.1 Establish traffic profile

The periodic behavior in cross-traffic is described by the time parameter T_p and the duration of burst/idle periods are described by the time parameters T_b and T_i as shown in Fig. 5.

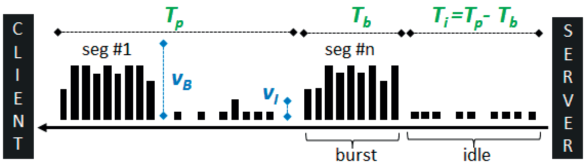


Fig. 5 Burst and idle periods in cross-traffic

In our earlier work [12] we described and analyzed a method for estimating the relevant cross-traffic parameters by using active packet pair probing. This method is based on that the probe packet receiver observes the inter-arrival time IAT) between probe packet pairs, compares it with the cumulative average IAT and filter the samples based on this.

If the sample is above the cumulative average, it is kept – if it is below, it is set to the average. The resulting time series of IAT observations ($t_{out,i}$) is used as input to computation of serial correlation up to a certain lag s (cf. Eq. 4). In our case where the packet pair probing period is known, the lag value maps directly over to the time dimension.

$$X_s = \frac{\sum_{i=1}^{N-s}(t_{out,i} - \overline{t_{out}})(t_{out,i+s} - \overline{t_{out}})}{\sum_{i=1}^{N}\left(t_{out,i} - \overline{t_{out}}\right)^2} \qquad (4)$$

In order to understand what the output of the serial correlation X_s tells us, a graphical view is recommended. By using this, the parameters of interest (T_p, T_b) are possible to identify (cf. Fig. 6).

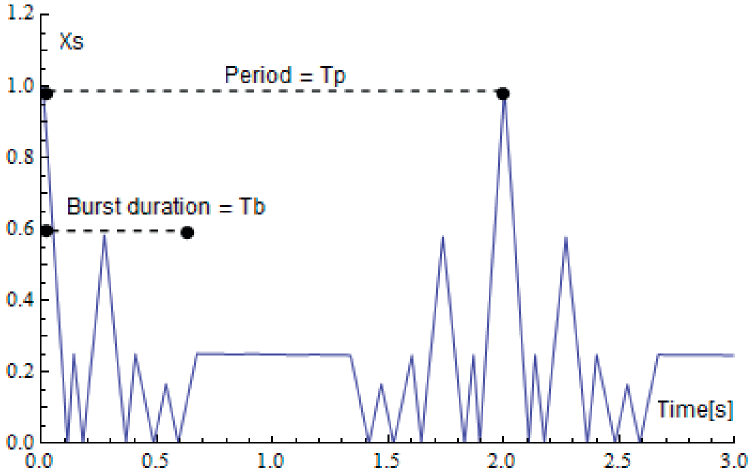


Fig. 6 Serial Correlation output for theoretical signal

There are several things one could read out from a serial correlation performed over a time series with a periodical component. The example X_s output shown in Fig. 6 is based on a theoretical signal with a period of 2s, and where the signal inside this period has an idle part of 1.3s and a burst part of 0.7s. The burst part is not a square pulse.

The presence of peak values are indicators of periodic components, and visible side lobes around a peak is an indicator of that the periodic component is not a perfect square pulse, but rather composed of several pieces. By looking at the lower range in time for the X_s output, the width (or duration) of the burst part for each period is visible.

The amount of probe traffic required in this phase is very low. In our earlier work we showed that accurate estimates for both T_p and T_b were possible to obtain using probe packet rates (f_p) down to 160pps. With a probe packet size of 100bytes it corresponded to a bitrate of 128Kbps.

4.2 Adjust probing to strata and synchronize

The direct output from the previous phase in terms of estimators for T_p and T_b are used in this phase in order to adjust the probing traffic according to the burst and idle periods. However, before the probe traffic is changed it is required to obtain some timing information which can be used to synchronize probing strata with the burst/idle periods in the cross-traffic.

The method we used for this purpose requires the presence of sequence number attached to each probe packet, and also the ability to restart the sequence numbering after a specific period.

Both these capabilities were supported by the probe traffic generator we used [15]. With this in place, the probe traffic used in the previous phase was changed so that it restarted its sequence numbers after the estimated T_p units of time, but keeping the same amount of probe packets per time unit (f_p). The probe packet sequence numbers were then available for use as an index (j), making it possible to summarize IAT observations ($t_{out,i,j}$) occurring at a specific time within each period (cf. Eq. 5) over n periods.

$$T_{out,n,j} = \sum_{i=1}^{n} t_{out,i,j} \qquad (5)$$

Further on, when performing this summarization for the whole index range N, it gives a the list $T_{Out,n}$ (cf. Eq. 6).

$$\begin{gathered} T_{Out,n} = \{T_{out,n,j}, 1 \leq j \leq N)\} \\ N = f_p * T_p \end{gathered} \qquad (6)$$

The purpose of producing the $T_{Out,n}$ list is to see where within the probe sequence of length N the burst period starts. The required number of periods n for which the summarization is required performed in order to give this type of information is not obvious. One might think that a high n value is good, but as the results will show in Section X this is not entirely correct.

Having identified the time within a probe sequence where the burst period starts, we have also established a timing reference between our probing and the bounds for burst and idle periods in the cross-traffic. These bounds can then be used to implement the stratified probing, where we change the active probing from the continuous f_p rate over to $f_{p,b}$ during the cross-traffic burst period and $f_{p,i}$ during the idle period.

4.3 Active probing and result collection

Choosing the optimal $f_{p,b}$ and $f_{p,i}$ values was outside of the scope for the research documented in this paper, and was left for future work. Thus, for the purpose of our experiments we chose a reference probe packet rate $f_{p,b}$=309pps for the burst period and set the $f_{p,i}$ to zero. The latter would not be appropriate in a scenario with a more complex cross-traffic mix than in our case, but sufficient to support the focus of this paper. The reference probe packet rate was used in the scenarios where the access capacity was at 10Mbps, independent of which quality level the adaptive video streaming (i.e. the cross-traffic service) was operating at. For the other access capacity levels, the $f_{p,b}$ was scaled up according to the changes in detected T_b. In other words, as T_b decreases (as a result of increased access capacity), $f_{p,b}$ was increased inversely proportional to this. This approach of scaling $f_{p,b}$ gave a constant number of probes during the burst period across all access capacity levels.

The probing pattern used in this phase differs slightly from the one used for establishing the cross-traffic profile. While in that phase we used packet pairs [12] with a certain gap between each pair, we use a continuous packet train (cf. Fig. 7) in this phase and collect IAT observations continuously. The reason for this change was to make better use of the information available in a probe packet sequence.

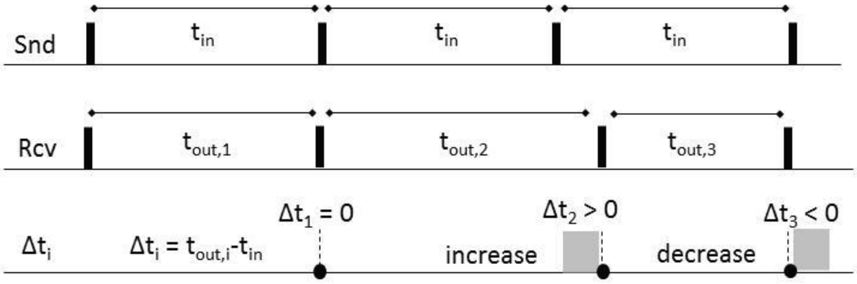


Fig. 7 Probe packet IAT observations

As illustrated in Fig. 7, the Δt_i values represent the difference in spacing between probe packets as sent and received. Depending how the cross-traffic impacts the probe packets Δt_i can be positive or negative. An increase in spacing between probe packets is a certain indicator of cross-traffic impact, but even a reduced spacing observation may contain cross-traffic impact information. The required processing in order to extract all information available in the IAT observations ($t_{out,i}$) will follow in the next section.

4.4 Calculate estimator for available bandwidth

For each $t_{out,i}$ observation, a cross-traffic delay component d_i is calculated and also a time shift element s_i. The latter is required in order for the subsequent calculation to be correct as a $d_i > 0$ would mean that the starting point for $t_{out,i+1}$ must be shifted. The continuous calculations are summarized as followed.

$$\begin{array}{lll}
if\ \Delta t_i = 0\ \&\ s_{i-1} = 0 & \text{then}\ d_i = 0, s_i = d_i & \\
if\ \Delta t_i = 0\ \&\ s_{i-1} > 0 & \text{then}\ d_i = s_{i-1},\ s_i = d_i & \\
if\ \Delta t_i > 0 & \text{then}\ d_i = \Delta t_i, s_i = d_i & (7) \\
if\ \Delta t_i < 0\ \&\ s_{i-1} > 0 & \text{then}\ d_i = [\Delta t_i + s_{i-1}]^+ & \\
 & \qquad S_i = [d_i - s_{i-1}]^+ &
\end{array}$$

Based on each d_i sample, a v_j sample can be calculated (cf. Eq. 8). This is used as input to the calculation of the total cross-traffic V_i (cf. Eq. 3) for a specific period. As per Fig. 5, the cross-traffic for the burst period is denoted V_B and for the idle period V_I. Since we are not probing during the idle period it gives that V_i=V_B.

$$v_j = C * d_i \qquad (8)$$

The estimator for available bandwidth $B_{i,Tp}$ during period i can then be calculated according to Eq. 3. In the results section we will also present the estimated burst rate $R_{i,burst}$ (cf. Eq. 9).

$$R_{i,burst} = \frac{1}{T_b} V_i[t - T_p, t] \qquad (9)$$

The reason why we also compute the burst rate is that we are interested in seeing how well the scaling of probe packet rate to $f_{p,b}$ based on burst period duration is performing. Alternative approaches for scaling it could have been based on access link capacity.

5. Measurement Setup

The purpose of the measurement testbed (cf. Fig. 8) is to provide both passive and active measurements of the cross-traffic generated across the access network. The passive measurements represents the real view and is collected by using TCPdump, while the active measurements represents the estimators obtained through stratified probing.

The testbed contains several components, ranging from the client side over to the server side. On the client side the video service is accessed by a dedicated Windows based PC, while the client receiving the probe traffic is on a separate Linux based PC. On the server side, we have the dedicated Microsoft Smooth Streaming server.

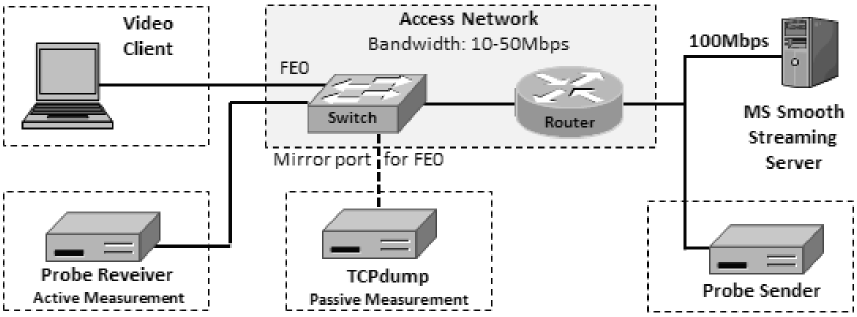


Fig. 8 Hybrid active/passive measurement testbed

The access network part consists of a Cisco 2960 switch and a Cisco 1800 model router. Using this type of commercial off-the shelf components gives access to useful functions [16] for controlling bandwidth similar to those used in commercial networks.

The probe traffic sender and receiver are based on the CRUDE/RUDE tool [15], which provides the sufficient capabilities for generating traffic pattern based on trace files.

When running the experiments under different conditions (access capacity, video stream level etc) we use a measurement period of 10 minutes in order to collect both passive and active measurement results. Including some automated post processing, one measurement round covering all the different access capacities (10-50Mbps, with increments of 5Mbps) for a specific cross-traffic profile would then typically last for about 2 hours.

6. Results

The results to be presented focus on demonstrating the capabilities of the method to support its four phases (cf. Fig. 4). The in depth analysis of each phase is covered in [12] and future work.

Typical results from each phase will be shown and explained. One specific scenario is being used through the following subsections and this is the case of a 5Mbps video stream as cross-traffic, configured with a segment request interval of 2sec and the access capacity set to 50Mbps.

6.1 Estimated traffic profile

The active probing using in this phase was based on packet pairs. Each packet had a size of 100byte. The time between paired packets was 0.5ms and the time between packet pairs was 6.1ms. This gave a probe packet rate of about 300pps (240Kbps), which then produced 150 probe samples per second. The output from the serial correlation is shown in Fig. 9 where the lag parameter has been converted to time dimension. The adaptive video stream representing the cross-traffic in this case was operating at a 5Mbps quality level and a segment fetch interval of 2sec, across an access capacity of 50Mbps.

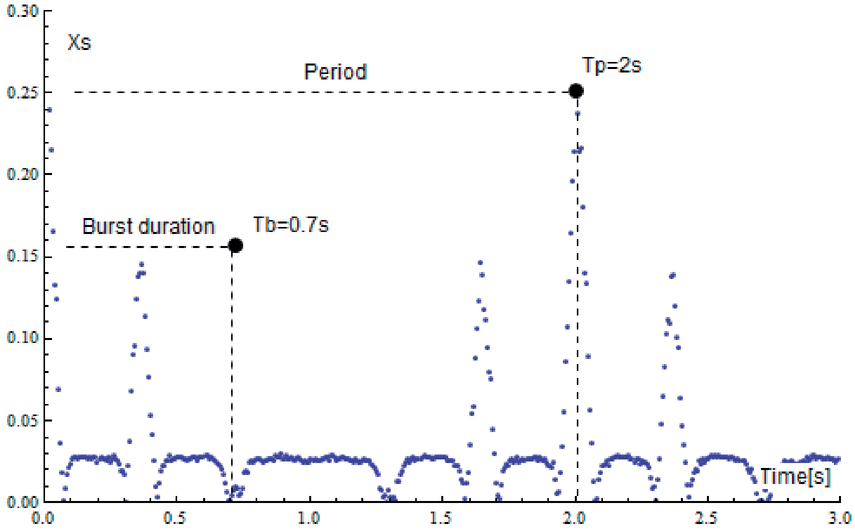


Fig. 9 Period and burst

We can see how the peak value in the serial correlation output X_s is able to detect the period of 2sec in the cross traffic, which is correct. The observed side lobes are quite similar to the theoretical output as shown earlier in Fig. 6. The burst duration is also visible as the point where X_s goes to zero after the last side lobe in the lower end of the time scale at about 0.7s. This matches the passive measured burst duration as shown in Fig. 2.

As described in [12] it is easier to read out the burst duration from the serial correlation output when the duration is low (e.g. at 100Mbps access capacity). In the cases studied for access capacities between 10-50Mbps, a combination of serial correlation output and the method applied in the next phase of the method would be beneficial.

6.2 Synchronization of probing strata with traffic profile

In this phase we used the same active probing as in the previous phase, but now re-configured so that the sequence numbering of the probes are restarted after 600 probe packets corresponding to the estimated period T_p=2s. The synchronization point we are looking for is the offset into the sequence of 600 probe packets where the burst period starts.

As described in the method section, the required number of rounds n required for the $T_{Out,n}$ time series to give useful output was not obvious. Therefore, we have shown the output of this calculation for different n values in Fig. 10.

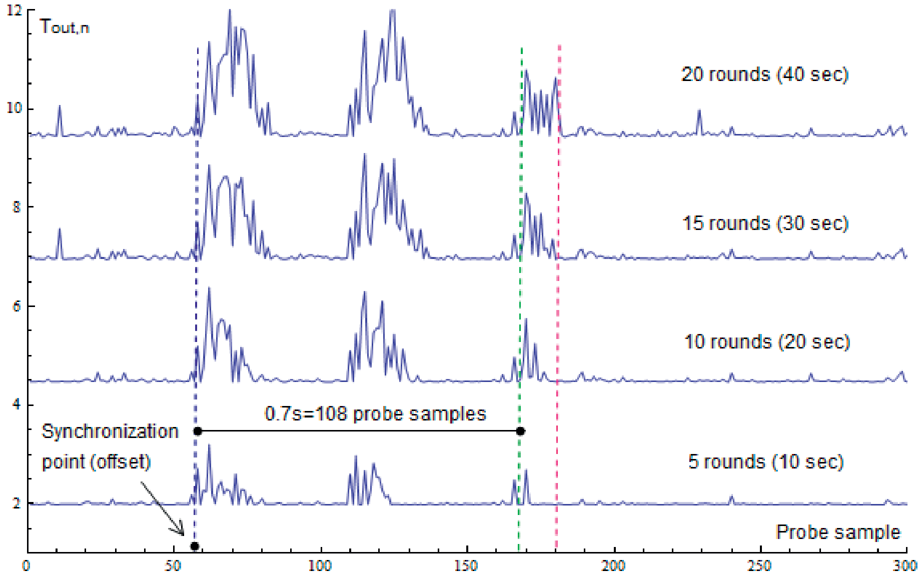


Fig. 10 Synchronization of probe strata and burst period

As we see, the case when using the lowest *n* value of 5 which corresponds to sample size of 10sec is the one which gives the narrowest view of the burst duration. The time for the burst start is the same for all n values, but as *n* increases the period of interest is stretched. Thus, for the purpose of synchronizing our probing into different strata (burst and idle), all values of *n* gives the same starting point for the burst strata. However, for the purpose of providing an additional view on the burst duration – in order to simplify the interpretation of the serial correlation output, the lower *n* values are better.

6.3 Probe traffic scaled by burst duration

In this phase the probe traffic was reconfigured from the packet pairs used in the previous phases, to a sequence of probe packet equally spaced with parameter t_{in}. Since finding the optimal probing level was outside of the scope for this research we choose a starting point based on some basic experiments. The starting point chosen was for a 10Mbps access, for which we considered 2.5% of the bandwidth as acceptable to make available for active probing. This corresponds to the first entry in Table I, where a t_{in} value of 3.24ms is given for all cross-traffic cases.

Table I Burst Period Probe Traffic

Access [Mbps]	Probe Packet Size [byte]	Probe Packet Spacing,t_{in} [ms]		
		3M video	4M video	5M video
10	100	3.24	3.24	3.24
15	100	3.06	3.08	3.00
20	100	2.92	2.86	2.62
25	100	2.79	2.43	2.50
30	100	2.20	2.24	2.36
35	100	1.89	2.11	1.97
40	100	1.80	1.99	1.81
45	100	1.71	1.74	1.71
50	100	1.66	1.37	1.59

The scaling of the probe traffic according to burst period duration was done manually in our experiment as per the passive measured values in Fig. 2. The 3Mbps video probing was scaled based on the T_b profiles for this specific quality level, and similar for both 4Mbps and 5Mbps video.

6.4 Estimated available bandwidth

The available bandwidth estimations for the different scenarios were done based on time intervals of 600 seconds with active probing of cross-traffic. The range of scenarios covered was all combinations 3/4/5Mbps video streaming as cross-traffic, across all access capacity levels from 10-50Mbps.

In order to assess the accuracy of the results obtained through the active measurements (probing), we first present the passive measured (by TCPdump) burst bitrates $R_{i,burst}$ (cf. Eq. 9) for the 5Mbps video stream across a 50Mbps access (cf. Fig. 11). The majority of the measurement samples are located around 14Mbps, but there is also a portion located around 25Mbps. The sample mean for the whole 600s period is at 16.2Mbps.

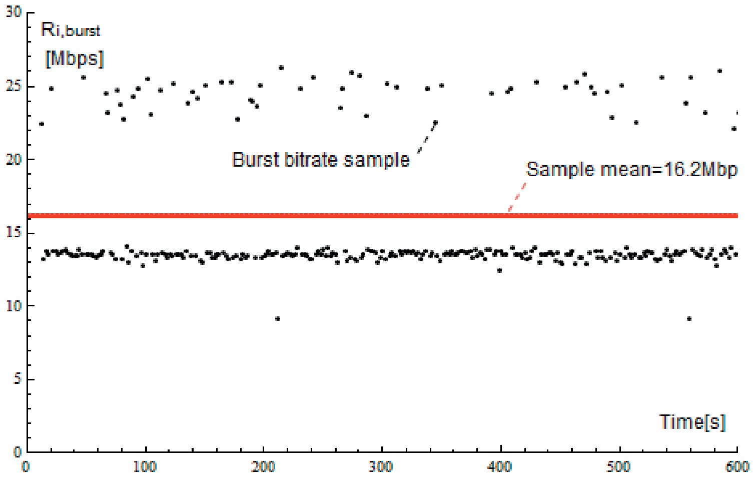


Fig. 11 Passive measured burst bitrates

Moving over to active measurements by means of our probing during the burst periods of the cross-traffic we got the results as illustrated in Fig. 12. The spread in the measurement samples here are higher than in the case of passive measurements, and we also notice that the sample mean is at 17.9Mbps. The included plot of a 30sec moving average for the measurement samples shows a significant reduction in distribution spread. This gives us an indication on how fast our method is able to come up with a reasonable accurate estimation of burst bitrate, and thus also available bandwidth.

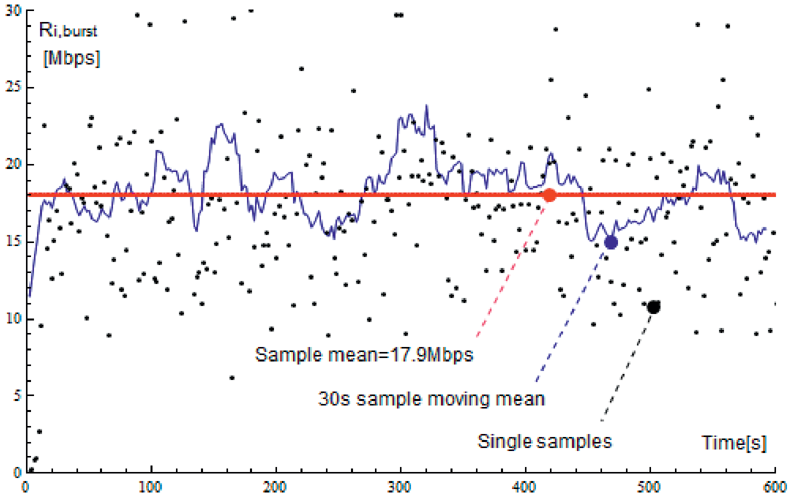


Fig. 12 Active measured burst bitrates

In the following we take a look at how well our method performs over a wider range of access capacities, but still for the same 5Mbps video stream (cf. Fig. 13).

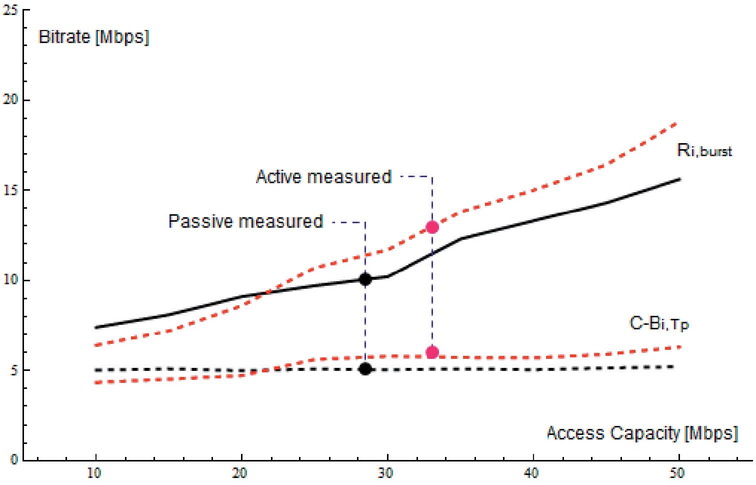


Fig. 13 Comparison of passive and active measurement

We see here that the specific case we have presented in detail (5Mbps video on 50Mbps access) is actually the worst result for all capacity levels. For all other access capacity levels the difference between passive and active measured $R_{i,burst}$ and subsequently $B_{i,Tp}$ is smaller. For the purpose of making the illustration better the plot shows $C\text{-}B_{i,Tp}$ (estimated cross-traffic) rather than $B_{i,Tp}$ (estimated available bandwidth).

The differences between active and passive measurements are at worst in the order of 20% when looking at sample means over the 600s period. However, this should not be considered as a real measure of the accuracy of the method, but rather just a starting point. With more effort put into finding optimal probing patterns and rates, the accuracy is likely to improve further.

7. Discussions

In our work we have made no attempt to compare the accuracy of our method against others. The main reason for this is that the published results for other methods and tools have not been addressing access links with burst traffic. Further on, those tools where source code are public available it would not be fair to test them for the sake of accuracy comparison in our scenario, as they were not made for this. It is also worth mentioning that most of these tools are about ten years old. However, it is worth mentioning that our stratified probing approach enables us to maintain a constant low probing rate even if the degree of burst duration is decreasing. This is the direct result of targeting the probes according to strata. In the scenario with 3Mbps video a cross-traffic the $f_{p,b}$ increases from the starting point of 309pps (247Kbps) up to 602pps (482Kbps), as the burst duration decreases from 0.72s to 0.37s. Looking at the f_p for the whole period T_p, remembering that $f_{p,i}$ is zero it is kept constant at 111pps (89Kbps) across all access capacity levels. The similar f_p values for respectively 4Mbps and 5Mbps video as cross-traffic are 160pps (128Kbps) and 210pps (168Kbps). We believe this clearly demonstrate the benefits of the stratified approach as we get more value for each probe sent.

In our work we have not used any specialized hardware to either generate traffic or to analyze it. All of the software used operates in user space and not kernel space of the operating systems. This directly implies that there is room for some errors in the results due to components performing multitasking during our experiments. We have tried to minimize the chances of this by following the advices found in [17] and also using dedicated nodes for each function in the experiments (cf. Fig. 8).

Our experimental approach to study our suggested method of stratified probing did not aim at developing a self-contained solution, which could be used outside a lab environment. However, there is a potential of doing so but it would require a certain amount of additional coding in the appropriate languages. This is outlined as part of future work, but it should be kept in mind that there are remaining technical challenges to be solved before it is recommended to invest this time. The main challenge in our view is to find a way to automatically choose a good starting point for the probe rates. In our work we used a level based on what we thought would be acceptable, but this assessment is of course highly subjective.

Even though we refer to our work as a new method for estimating available bandwidth on access links, we acknowledge that there are a lot of scenarios which are not covered by our approach. An example of this would be cases where periodic components in cross-traffic are not present at all. In such cases, our method does not add any value. In light of this, our method could be considered as a candidate component to be included in other more general methods.

8. Conclusions

The results from our empirical evaluation of the suggested method of applying a stratified approach for probing of cross-traffic are quite promising. We have showed that the different phases of our method are possible to implement when using a specific service type as cross-traffic. The choice of adaptive video streaming as the cross-traffic makes the findings quite relevant, reason being the growing amount of services with video components on the Internet.

The approach of using serial correlation as for analyzing time series of observations, such as IAT observations between probe packets is quite powerful. In this area we only provided a brief introduction to the concept, as we have presented this part in more detail earlier [12].

The benefit of stratifying the probe traffic according to the cross-traffic profile is quite clear. We have showed that our method is able to maintain about the same level of accuracy in the available bandwidth estimates over a wide range of burst degrees.

In order for our method to apply, there must be periodic behavior in the cross-traffic. This will not always be the case, but as the popularity of video services is growing we believe it will be quite common to see such behavior, especially on access links serving residential users.

A more complex traffic mix may of course change some of our results, but we believe that future methods in this domain should attempt to use the presence of periodic cross-traffic to their benefit in terms of improving accuracy in available bandwidth estimations.

9. Future Work

The use of a more complex cross-traffic is interesting to study, in order to see how well our approach would perform in such a scenario. From one perspective it may add more complexity to the different phases of our method, but it may also contribute to smooth out sub-bursts and thereby simplify detection of burst durations. A burst period which is closer to a square pulse profile gives a clearer output when subject to serial correlation.

In order to further justify the gain of using knowledge about periodic traffic components as part of available bandwidth estimations, it would also be beneficial to collect and analyze traffic on a more aggregated level from real access networks.

Finding optimal probing patterns and rates was outside of the scope for our work. In order to take the validation of the method potentially one step further, we believe this would be an important area to investigate. We are especially interested in using single packet delay observations as basis for the active probing. Reason being that it could simplify the processing of observations on the receiver side, and also make the probing less vulnerable for packet loss and out-of-sequence events.

References

[1] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," Multimedia, IEEE, vol. 18, no. 4, pp. 62 –67, april 2011.

[2] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," Network, IEEE, vol. 17, no. 6, pp. 27 – 35, nov.-dec. 2003.

[3] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, ser. IMC '03. New York, NY, USA: ACM, 2003, pp. 39–44.

[4] N. Hu, S. Member, P. Steenkiste, and S. Member, "Evaluation and characterization of available bandwidth probing techniques," IEEE Journal on Selected Areas in Communications, vol. 21, pp. 879–894, 2003.

[5] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, "Multifractal cross-traffic estimation," in Proc. ITC Specialist Seminar on IP Measurement, Modeling, and Management, 2000, pp. 15–1.

[6] E. Goldoni and M. Schivi, "End-to-end available bandwidth estimation tools, an experimental comparison," in Traffic Monitoring and Analysis, ser. Lecture Notes in Computer Science, F. Ricciato, M. Mellia, and E. Biersack, Eds. Springer Berlin Heidelberg, 2010, vol. 6003, pp. 171–182.

[7] G. Urvoy Keller, T. En Najjary, and A. Sorniotti, "Operational comparison of available bandwidth estimation tools," ACM SIGCOMM Computer Communications Review, Volume 38 N°1, January 2008, 01 2008.

[8] J. Sommers, P. Barford, and W. Willinger, "Laboratory-based calibration of available bandwidth estimation tools," Microprocess. Microsyst., vol. 31, no. 4, pp. 222–235, 2007.

[9] B. J. Villa and P. E. Heegaard, "Improving perceived fairness and qoe for adaptive video streams," ICNS 2012, March 2012.

[10] B. J. Villa, P. E. Heegaard, and A. Instefjord, "Improving fairness for adaptive http video streaming," in EUNICE, 2012, pp. 183–193.

[11] B. J. Villa and P. E. Heegaard, "Group based traffic shaping for adaptive http video streaming," in Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013), March 2013.

[12] B. J. Villa and P. E. Heegaard, "Detecting period and burst durations in video streaming by means of active probing," in Proceedings of the 3rd International Conference on Computer Communication and Management (ICCCM 2013).

[13] A. Rizk, Z. Bozakov, and M. Fidler, "H-probe: Estimating traffic correlations from sampling and active network probing," CoRR, vol. abs/1208.2870, 2012.

[14] A. Zambelli, "IIS smooth streaming technical overview," http://www.microsoft.com/silverlight/whitepapers/, Tech. Rep., March 2009.

[15] J. Laine, S. Saaristo, and R. Prior, "Real-time udp data emitter (rude) and collector for rude (crude)," http://rude.sourceforge.net/, Jan 2000.

[16] C. H. Tim Szigeti, End-to-End QoS Network Desi. Cisco Press, November 2004.

[17] R. Prasad, M. Jain, and C. Dovrolis, "Effects of interrupt coalescence on network measurements," in In Proceedings of PAM '04, 2004.

PAPER 11: Investigating Quality of Experience in the context of adaptive video streaming: findings from an experimental user study

Bjørn J. Villa, Katrien De Moor, Poul E. Heegaard, Anders Instefjord

Published in
Proceedings of Norsk Informatikkonferanse 2013 (NIK 2013)
Tapir Akademiske Forlag, ISBN 978-82-321-0365-2 . pp. 122-133
Nov 18-20, 2013, Stavanger, Norway

Investigating Quality of Experience in the context of adaptive video streaming: findings from an experimental user study

Bjørn J. Villa, Katrien De Moor, Poul E. Heegaard, Anders Instefjord

Abstract: Although adaptive video streaming solutions have become very popular over the last years, only a limited number of studies so far have investigated Quality of Experience (QoE) in the context of such services from a real user perspective. In this paper, we present results from a user study (N=32) on adaptive video streaming QoE. Content (i.e., music video clips) was streamed on portable terminal devices (namely iPads), using different bitrate profiles which represent realistic bandwidth conditions. All users conducted the test in two different usage scenarios, representing different social settings. QoE was evaluated in terms of traditional QoE-measures and complemented with a number of alternative, affective-state related measures. The findings indicate differences between the considered bitrate profiles in terms of QoE (traditional and alternative measures) and indicate that a lower, constant bitrate seems to be preferred over the changing bitrate which is - on average - higher. This is the case for both separate usage scenarios. Although we found no significant influence of the social setting when considering traditional QoE evaluation measures, some significant differences between both settings were detected when evaluating QoE in terms of delight and annoyance.

1. Introduction

Internet traffic has tremendously increased over the past years and is expected to still increase four times in the period 2010-2015 [1]. Forecasts indicate that by 2015, consumer Internet traffic will be dominated (growing to over 60%) by Internet video and that video in its different forms will represent around 90% of the global consumer traffic. For mobile traffic, the estimated volume of video services amounts up to around 90% in 2015 [2]. This evolution has been driven by the availability of broadband access at affordable prices, in parallel with new types of end user equipment. It also has important implications for the delivery of services to end users. The opportunity to provide advanced services as part of the best-effort Internet traffic class has been utilized by many successful service providers, such as YouTube and Netflix. This type of service delivery is commonly known as the Over-The-Top (OTT) model [3], where any service provider can provide services to any customer – on a global basis.

The OTT service delivery model provides opportunities, but also introduces technical challenges. The best-effort Internet traffic class gives no guarantees in terms of QoS metrics such as e.g., bandwidth. In order for services to perform well under such conditions, the concept of adaptive services has emerged. These services have the ability to change their requirements during service delivery according to experienced network characteristics. Looking at video services in particular, the concept of Dynamic Adaptive Streaming over HTTP (DASH) has been the subject of recent standardization efforts [4], and is already in use by major content providers. It has also been argued that adaptive streaming will become even more important in the future: Oyman and Singh [2] refer to a recent study by TDG Research, which forecasted that by 2015, 51% of Internet video will be supported by adaptive streaming solutions. Further on, the results from a survey conducted by the DASH Industry Forum among European Broadcasters [5], show that the majority of them will deploy DASH solutions during 2013 and 2014.

Despite the increasing popularity of DASH based services and its assumed capability to providing better QoE, only a limited number of studies so far have investigated this topic from a real user perspective. This means e.g., considering specific factors that may have an influence, and considering different usage scenarios. In this paper, we therefore aim to 1) investigate QoE in the context of different adaptive video streaming profiles, and 2) explore the possible influence of social setting on QoE in this respect. To this end, we present results from an experimental user study in which three recent music video clips were streamed on iPads as portable terminal devices, and evaluated in terms of QoE by 32 test subjects. Since bandwidth is still seen as one of the most important bottlenecks [2], three different bitrate conditions were used. These conditions correspond with three distinct, realistic bandwidth and thus video delivery profiles (i.e., constant high bitrate, constant low bitrate, changing bitrate from high to low). Two different usage scenarios, representing different social settings (i.e., alone, dedicated internet access vs. together with others, notion of sharing internet access with others), were investigated in the test. QoE itself was evaluated in terms of traditional QoE-measures, complemented by a set of alternative, affective-state related measures. The latter were included to link up to the new definition of QoE as 'a degree of delight or annoyance …' proposed in [6].

The rest of this paper is organized as follows: section 2 discusses the concepts of adaptive video streaming and Quality of Experience in more detail and give a brief overview of relevant related work. Thereupon, Section 3 discusses the details of the experimental study set-up (i.e., sample description, test procedure, network, service and device configuration, test material and test environment, the usage scenarios and included subjective measures). Section 4 is dedicated to the analysis and discusses the results of the study. Finally, Section 5 shares our conclusions and suggestions for future research.

2. Related Work

Whereas QoE has been defined by ITU [7] as the ‘overall acceptability of an application or service, as perceived subjectively by the end-user’, this dominant definition has been criticized over the last years due to its narrow interpretation (see [8] for an overview). In 2012, a new definition of QoE as ‘the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user’s personality and current state’ [6] was proposed. The goal is to bring users in the ‘zone of delight’ and to prevent/reduce user annoyance. In addition, a range of human-, system- and context-related factors that may bear an influence on QoE, are mentioned

[6], implying that QoE evaluation requires an interdisciplinary approach (as is the case in our work). This more holistic approach to QoE was developed as part of a bottom-up and community-driven process in the COST Qualinet project on QoE. However, an important challenge is still to translate this broader conceptual understanding of QoE into adequate and extended measurement approaches, which also consider alternative measures of QoE (as indicators of 'delight' or 'annoyance') and which draw on interdisciplinary collaboration. Although some recent studies have taken some first steps to confront the traditional QoE measures with alternative self-report measures (see e.g., [9]), this is still largely unexplored territory. By including both traditional and alternative self-report measures of QoE (see Section 3), the study presented in this paper aims to make a relevant contribution in this respect.

Since DASH solutions are based on the principle of adjusting the quality level on a continuous basis during delivery [4], they imply a need to develop adjusted, reliable QoE evaluation methods, as is argued in [2]. In their work, an overview is given of a set of 'QoE metrics' for DASH. These include e.g., average throughput, initial playout delay and buffer level. Although a number of very pertinent integration challenges with respect to QoE optimization for DASH services are discussed and although a QoE evaluation approach is proposed, an important limitation of this work is that QoE issues are not tackled from a real user point of view (i.e., QoE as inherently subjective). QoE is not considered from a more holistic perspective, thus leaving a lot of questions unanswered. In [10], a novel approach to estimate QoE for adaptive HTTP/TCP video streaming is introduced. The proposed no-reference QoE estimation module differs from existing approaches: both video playout interruptions - which may occur due to fluctuations in the bandwidth and retransmissions of packets - and the quantization parameter is considered. The latter indicates the video compression and is therefore directly linked to the bitrate and possible changes in the video quality [10]. This estimation approach is valuable, yet, it also approaches QoE from a very narrow, technical point of view and the output is a prediction of 'isolated' QoE in terms of MOS-values.

One of the few papers presenting results from two actual subjective video quality experiments in the context of HTTP adaptive streaming, is the work by Robinson et al. [11], which we will briefly discuss below. They have tested several commercially available solutions for HTTP adaptive video streaming, including Apple HTTP Live Streaming (HLS)[12], Microsoft Smooth Streaming [13] and Adobe Dynamic Streaming [14]. Although these solutions are quite similar in terms of how they operate, they are not interoperable. The main reasons for this are the differences in encoding and client/server message format. The ongoing standardization efforts [15,13] are addressing these issues. In addition to the differences found on the client/server interface, there are also differences in internal functionality on the client side, which impacts the behavior of each solution. As a result, the algorithms controlling when to change quality level, and how often these changes are allowed are different between the solutions. The latter is either decided by video segment duration- or volume. Whereas the segment duration approach is applied in the commercial available solutions from Microsoft, Apple and Adobe, the segment volume based approach is e.g. used in the closed solution used by Netflix [16]. As is discussed in more detail in section 3.3, we used Apple HLS in our study.

The first study of Robinson et al. [11], in which these commercial solutions were used, took place in a controlled lab environment and investigated the impact of different network-related impairments on overall video quality (MOS scores). The results indicate high scores for the three commercial solutions when no impairments are introduced, yet point to a number of differences – which can be linked to the distinct characteristics of the different solutions – when considering the impact of the investigated impairments (namely bandwidth limitations, latency, random and burst packet loss). Moreover, relatively high MOS scores were found up to 2 Mb/s. For the second study, another approach was followed: members from the Alcatel-Lucent youth lab were invited to participate in an online study in which clips containing typical yet exaggerated impairments in HAS were evaluated in terms of overall acceptability (transformed into MOS scores). The results indicated amongst others, that a constant (or approximately constant) bit rate is preferred over a bit rate that is frequently changing, even when this implies that the average quality is lower [11].

Both the Robinson et al. studies and the work by Singh et al. [10] are valuable contributions. However, QoE is only evaluated in terms of overall video quality perception. In our own study, we aim to extend this traditional approach, by linking up more explicitly to the new definition of QoE and by introducing alternative measures to evaluate QoE for adaptive video streaming. In the next section, we discuss the setup of the study in detail.

3. Experimental study setup

As mentioned, the objective of our study is twofold. First of all, we aim to investigate QoE in the context of different adaptive video streaming conditions. To this end, three different bitrate profiles, representing realistic bandwidth conditions (i.e., constant high, constant low and changing), were used in the experiment. A second objective is to explore the possible influence of specific contextual variables on QoE. More concretely, two usage scenarios were considered in the test representing different social settings (i.e., alone, dedicated internet access vs. together with others, notion of sharing internet access with others).

3.1 Sample description

Based on the widely used convenience sampling method and following the ITU guidelines concerning the number of observers for subjective quality assessments (i.e., between 15 and 40), 32 test subjects were recruited. The main disadvantage of this type of sampling lies in the fact that the results cannot be generalized to the general population. However, as this is not the aim of our study, the use of this sampling method is justified. 59.4% male and 40.6% female test subjects, participated in the main study (in groups of four). The average age of the participants was 22 (Standard Deviation (S.D.)= 4.6). The youngest participant was 17, the oldest 33. All test subjects had normal or corrected-to-normal vision and reported normal hearing. Whereas a limited number of participants were working or studying in the field of audio/video quality, multimedia processing or a related field, the majority (81.3%) were naive users. Next to basic socio-demographical data, we also collected some additional information about the participants. In particular, we were interested in their use or non-use of tablets in their natural environment. 40.6% owns a tablet and one out of three participants used a tablet on a daily basis or even several times per day during the month before the test (which was used as reference period in the questionnaire). Moreover, 68.8% of the participants watched video content on a tablet during the month preceding the study. In order to link up to the new QoE definition, which also emphasizes the importance of a user's 'current state', we also included a pictorial measure of 'mood' (the validated 'Pick-a-Mood'-scale) in the pre-questionnaire [17]. Whereas 4 respondents (12.5%) evaluated their mood as 'neutral', 81.3% of the test subjects indicated to be in a pleasant state. Two participants labeled their mood as unpleasant (respectively 'tense' and 'bored').

3.2 Test procedure

The test procedure consisted of 3 parts and the entire test took around 1 hour. The main experiment was preceded by a pilot test, which was used for optimizing the procedure, the instructions and questionnaires. The procedure was as follows:

In the first part, the participants were given a brief, general introduction to the study (as in [11], without revealing concrete aims or details) and asked to fill in the pre-questionnaire. Thereupon, they were given instructions with respect to the viewing procedure and the operating of the iPad video player. The participants also received a summary of the instructions (with screenshots) on paper. Then, the self-report measures and different scales used in the during-questionnaires (see section 4.1) were explained. Finally, to familiarize the participants with the quality range and evaluation scale, they were shown examples of the worst and the best quality used in the test. Before moving to the next part, the test subjects could ask questions in case something was not clear.

After the introduction and briefing phase, the actual test started (part 2), with the first usage scenario. The participants were informed that they would watch three online music videos in the current setting, namely a situation where they are sitting alone at a desk and have their own, dedicated internet access point next to them. After the watching of a clip, the during-test questionnaire was filled in (considering the entire clip; so one rating per measure per clip) and this procedure was repeated 3 times.

Thereupon, the participants were asked to move to the second usage setting, reflecting a situation in which a user is alone, but watching in the presence of other people and sharing the internet access. Before the next clip viewing round started, the test leader pointed to the different situation (together with other people and sharing the internet access). Again, they watched the three clips, and filled in the during-questionnaire after every clip (considering the entire clip).

3.3 Sample description

We used 9.7" iPads (4th gen) as terminal devices in this study. This choice was made based on recent figures for the worldwide tablet market [18], which indicate that Apple is still the number one vendor in terms of market share. The participants were asked to respect an indicated viewing distance of 50 cm. During the playback of the videos, the participants were wearing Bose AE2 headphones, which were set to a medium, comfortable level. The participants were able to change the volume in case needed.

The use of iPads in the experimental part of our work made it necessary to use Apple HLS. The HLS solution is developed for the purpose of streaming audio and/or video from web servers to iOS based clients such as iPhone and iPad. The recommended segment duration value for HLS provided content is 10sec which then gives the highest possible frequency of quality adjustments. A change in quality level is requested by the client whenever it observes changes in available bandwidth, which makes it desirable to either go up or down in quality level. The details of the controlling algorithm on the client side are not disclosed by Apple. In order to make the experiments as realistic as possible, we established a complete network and service platform (cf. Fig. 1). The alternative would have been to use locally stored content on each iPad with impairments injected through the encoding process.

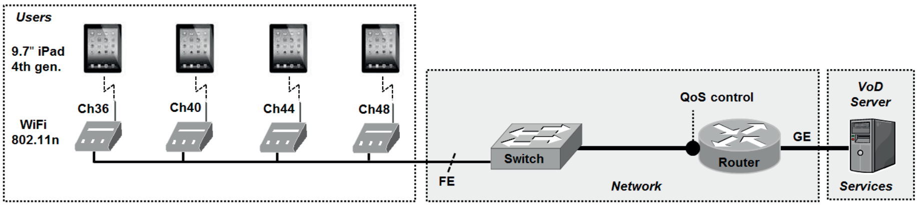


Figure 1 Lab network

Each of the iPad's are connected to the lab network through dedicated 802.11n access points, each of which was configured to use non-overlapping channels (36/40/44/48) to avoid interference. In order to provide QoS control for the video streams during the experiments, bitrate-limiting functions provided by the Cisco router in the network part were used. The QoS control was done individually for each iPad. The role of the QoS control function in the experiments was to enforce the delivery of video streams according to the configured profiles as illustrated in Fig. 2 on the left side. The right side of this figure shows how the actual streams behaved when being controlled by these profiles. As expected, there are some slight variations between the sessions. This is due to the natural lack of synchronization of the video start operation on each tablet. The initial behavior of any adaptive video streams is also visible, as it always starts out low and climbs until it reaches the maximum level possible given by the stream definition and current bandwidth constraint.

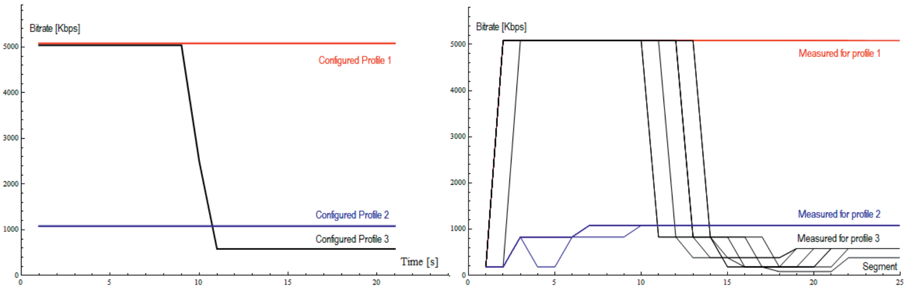


Figure 2 Configured (left) and measured (right) bitrate profiles for video streams

The three bitrate profiles represent different available bandwidth scenarios. The first profile is the case where the video stream is allowed to operate at maximum bitrate level (5Mbps). For the second profile the bandwidth is constrained so that the client decides to request a lower bitrate (1Mbps). In the third profile, a bandwidth constraint is introduced half way into the music video, which makes the client request a bitrate level from the lowest region (600Kbps). In none of the above profiles, the audio quality was affected.

3.4 Test material and test environment

The tests took place at the university campus, in a dedicated test room, with semi-artificial lighting. As test material, we used 3 recent music video clips of female singers, with duration of about 210sec. These clips were selected from a larger content pool, based on their relatively similar spatial and temporal information. Each clip was available in 14 different quality levels – encoded as MPEG-2 Constant Bitrate (CBR) transport streams. In Table 1 we present those levels actually used in the experiment (cf. Fig. 2, right side) in more detail. Depending on the spatial / temporal complexity of the video, the encoding process applies the appropriated degree of compression.

Table 1 Quality levels / bitrate profiles used in experiment

Bitrate (Kbps)	5000	1000	750	500	200	100
Resolution (Pixels/Inch)	95010	34204	23753	15202	5938	1518

The sequencing of the clips and order of the bandwidth limitations were presented in such a way that order or other presentation-related effects were avoided.

3.5 Usage scenarios

The creation of different Internet access- and social setting scenarios was done by running the experiments in groups of four users, as was already mentioned above. The first usage scenarios we created for the group members was the case where we let them sit alone and with their own access point visible on their desk (see Fig. 3, left). When placed in this scenario, information was given stating that they were now accessing the Internet through a dedicated access.

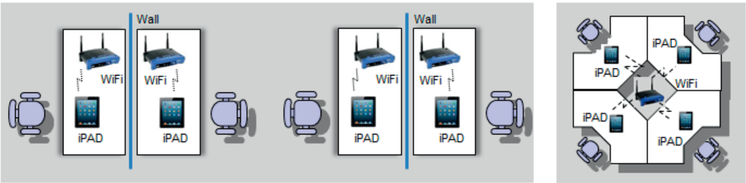


Figure 3 (left) Users separated with dedicated access; (right) Users together with "shared" access

In the second scenario, the idea is to create a notion among the users that they are part of a group which is sharing the access to Internet. This scenario was created by letting the group members sit at the same table where they could see each other, and with a single access point on the desk (Fig. 3, right). When placed in this scenario, information was given stating that they were now accessing the Internet through a shared access. The access point placed on the shared desk in this scenario was only powered on, but did not provide service to the iPads. Put differently, whereas the participants were given the impression that they were sharing the bandwidth, each iPad was still connected to the lab network through the dedicated access points.

3.6 Subjective measures

Before turning to the results, Table 2 overviews the subjective measures included in the questionnaires after each clip, relevant for this paper.

Table 2 Overview of used subjective measures

Overall video quality	ITU five-point Absolute Category Rating (ACR) scale (due to content length, a double stimulus presentation design was not appropriate)
Acceptability	binary scale (acceptable / not acceptable)
Degree of delight and *degree of annoyance*	5-point scale with labeled end points, ranging from 0 ('not at all', e.g., feeling not annoyed or not happy at all) to 4 ('extremely', e.g., feeling extremely annoyed or happy). Multiple item constructs, so internal consistency reliability was checked by means of Cronbach's alpha (threshold of .65). New variables were computed for delight (α= .89, 3 items) and annoyance (α= .87, 2 items)
Pleasure and *arousal*	Nine-point SAM-scales, ranging from extremely sad to extremely happy (pleasure dimension) and from extremely calm/bored to extremely excited (arousal dimension). Self-Assessment Manikin (SAM) is a pictorial technique for assessing specific emotions in relation to a stimulus
Expectations	Degree to which the overall video quality was meeting the participant's expectations: measured on a 5-point scale ranging from -2 ('much worse than I expected') to 2 (Much better than I expected')
Perceived video quality stability	5-point scale, based on two items (α= .88)

4. Results

4.1 QoE evaluation

We first of all take a look at the figures for the 'traditional' QoE measures included in the study, namely the overall quality ratings and acceptability evaluations. As the descriptives (Average, Avg. and Standard Deviation, S.D.) in Table 3 indicate, the overall quality of the videos is on average evaluated as best in the constant high bitrate profile and as worst in the profile in which the bitrate changed from high to low. Moreover, this is the case in both usage scenarios (Avg. = 4.44 vs. 3.00 in scenario 1, Avg. = 4.34 vs. 2.78 in scenario 2).

Table 3: Descriptives (Avg. & S.D.) of subjective measures, per scenario and bitrate profile

Usage scenario	Scenario 1 (alone, dedicated access)						Scenario 2 (together, shared access)					
Bitrate profile	high		changing		low		high		changing		low	
	Avg.	S.D.	Avg.	S.D.	Avg.	S.D.	Avg.	S.D.	Avg.	S.D.	Avg.	S.D.
Overall quality	4.44	0.88	3.00	0.95	3.75	0.67	4.34	0.60	2.78	0.83	3.59	0.80
Delight	2.46	0.90	1.51	0.79	2.00	0.96	1.74	0.95	1.41	0.81	1.61	0.96
Annoyance	0.22	0.44	0.92	0.88	0.38	0.79	0.45	0.77	1.23	1.03	0.58	0.64
Arousal	7.09	1.20	5.84	1.27	6.56	1.48	6.25	1.41	5.22	1.43	5.84	1.48
Pleasure	4.47	1.93	3.28	1.40	3.69	2.04	3.72	2.00	3.00	1.67	3.03	1.71
Expectations	0.75	0.95	-0.47	0.92	-0.13	0.83	0.59	0.76	-0.69	0.82	-0.06	0.56
Video quality stability	3.23	0.93	0.63	0.68	1.83	1.29	3.33	0.71	0.64	0.64	2.03	0.84

Figures 4a and 4b show absolute numbers and thus give a more nuanced overview of the distribution of the overall quality ratings. As can be observed, in both usage scenarios, the overall quality of the videos shown with high constant bitrate is clearly evaluated as better than the videos shown with a low, but constant bitrate and those with a changing bitrate. Thus, although the changing profile implies a bitrate that is on average higher, our findings indicate that a lower but constant bitrate scores higher in terms of overall quality. To investigate whether these differences depending on the bitrate profile are significant, non-parametric tests were used. For both usage scenarios, we conducted Friedman's ANOVA with the overall quality ratings as dependent and the different bitrate profiles as independents. Based on these tests, we have no evidence to state that the differences between all three bitrate profiles in terms of overall quality, as shown in figures 4a and 4b, are significant from a statistical point of view.

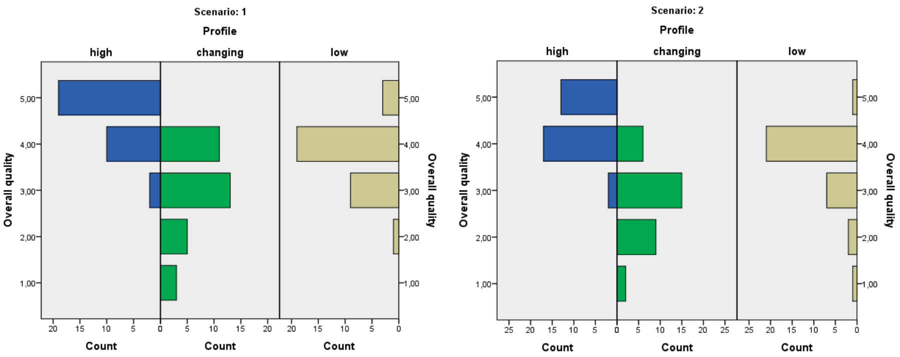


Figure 4a and 4b: absolute ratings for overall quality, for the bitrate profiles and usage scenarios

Purely by looking at the descriptives and figures, it can also be argued that the overall quality ratings for the different bitrate profiles are – in general - relatively high. This is illustrated in the boxplot for the overall quality ratings (Fig 5a): even for the changing bitrate profile, the median - as indicator of the middle tendency – is still relatively high (namely 3, indicating a 'fair' overall quality).

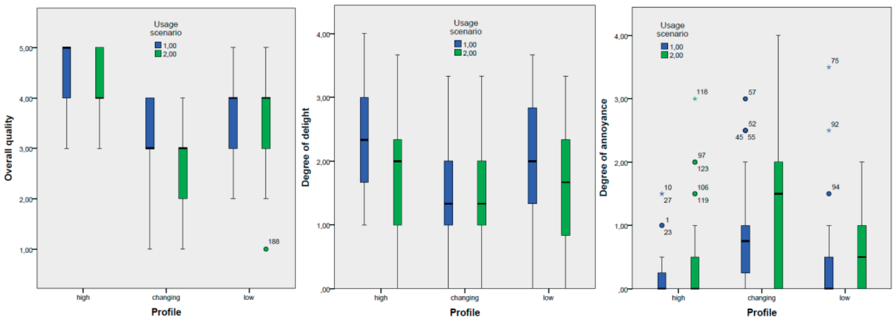


Figure 5a, 5b, 5c Boxplots for overall quality, degree of delight and degree of annoyance

When considering the acceptability of the overall video quality (i.e., whether or not it is good or good enough to watch the video) for the different bitrate profiles and scenarios, we found that only 7.3% of all videos were evaluated as ‘not acceptable’ in the scenario where participants were watching alone and had a dedicated network access (scenario 1). In the second usage scenario, this percentage is a bit higher, but still relatively low (14.6%). These figures indicate that – in both settings – the majority of the videos were evaluated as acceptable to watch, despite the fact that the test subjects noticed some interruptions and assessed the overall video quality differently according to the bitrate profile. Based on these observations, it can be argued that the strong fixation on the evaluations of overall video quality in terms of MOS scores as only measure of QoE, needs to be nuanced and broadened. It is however interesting to further explore which videos were considered as not-acceptable. In this respect, the acceptability ratings support the findings that were discussed above: the percentage of videos that are rated as not-acceptable, is highest within the changing bitrate profile (i.e., 71.4% in scenario 1 and 78.6% in scenario 2). We used Pearson’s chi-square test to investigate the relation between the evaluated acceptability and the three bitrate profiles and found that there is a significant correlation ($\chi^2(1)= 19.489$, $p < 0.001$) between the bitrate profile (changing vs. constant) and the acceptability of the overall video quality. This seems to represent the fact that, based on the odds ratio, the odds that a video was evaluated as acceptable were 8.1 times higher if the bitrate profile was constant instead of changing.

As was discussed above, we also included a set of ‘alternative’ measures of QoE in our study. The descriptives in table 2 allow us to share some general observations, before further investigating the differences across the different bitrate profiles. On average, the degree of delight (see Fig. 5b), the self-reported arousal and pleasure and the evaluated video quality stability were highest in the constant high bitrate profile and lowest in the changing bitrate profile. Moreover, this is the case in both usage scenarios. Similarly, on average, the overall quality of the clips shown in the high bitrate profile is slightly better than expected, whereas in the changing and low but constant bitrate profiles, the overall quality is evaluated as worse to slightly worse than expected, as is also illustrated in Fig. 6.

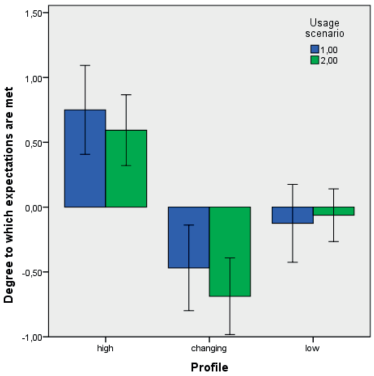


Figure 6 Avg .expectations (95% C.I.)

Table 4 Friedman's ANOVA results

	Usage scenario 1	Usage scenario 2
	sig.	*sig.*
Degree of delight	N.S.	N.S.
Degree of annoyance	N.S.	N.S.
Pleasure	< .05	N.S.
Arousal	.05	.05
Expectations	N.S.	N.S.
video quality stability	N.S.	N.S.

With respect to the degree of annoyance (see Fig. 5c), the opposite finding holds true: the annoyance is highest in changing bitrate profile and again, this applies for both scenarios. Again, to investigate whether these observed differences are substantial, Friedman's ANOVA tests were performed with the alternative measures as dependents and the different bitrate profiles as independents. The results, which are summarized in Table 4, point to significant differences between the three bitrate profiles, but only for reported Pleasure and Arousal. To further explore where the differences are situated, the post hoc procedure consisted of the conducting of additional separate Wilcoxon signed rank-tests. For the *first usage scenario*, the differences between the constant high bitrate profile and the changing bitrate profiles for the self-reported Pleasure and Arousal are substantial. The results also indicate a significant difference between the changing and constant low bitrate profile in terms of Pleasure. The reported degree of delight and video quality stability are higher in the constant low bitrate profile than in the changing profile, but we could not find statistical evidence supporting this observation. In the *second usage scenario* the arousal is significantly higher in the constant low bitrate profile than in the constant high bitrate profile.

4.2 Does social setting and internet access sharing influence QoE ?

As can be observed in Table 3, the subjective measures consistently point to a better QoE (e.g., better overall quality evaluation, higher delight, lower annoyance) in scenario 1. Additional analyses were performed to test whether there are significant differences between usage scenario 1 and 2 in terms of QoE (traditional and alternative measures) per profile. As we aim to compare two related, we used Wilcoxon signed-rank test, which can be considered as a non-parametric alternative for the dependent t-test. For the constant high bitrate profile, we found no significant differences in terms of QoE (using both the traditional and alternative measures) between the first and the second usage scenario. Or put differently, in this study, the presence of others and the notion of dedicated vs. shared access did not have a substantial influence on the QoE.

When investigating the differences in scores between the two scenarios for respectively the changing and the constant low bitrate profile however, we found significant differences (p< .05) in terms of the self-reported pleasure, arousal and the degree of delight, which were all significantly higher in the first usage scenario. This means that the differences between usage scenario 1 and 2 for the changing and the constant low bitrate profile, are substantial. Finally, the acceptability ratings (compared using Pearson's chi square test) did not significantly differ between the two usage scenarios, meaning that the presence of others and notion of shared internet access – as operationalized in our study - did not have a significant impact on the participants acceptability evaluations (they were more explicitly and significantly associated with the different bitrate profiles, as was discussed above).

Follow-up research with additional test subjects is needed to validate these findings and observations, as well as to fully understand their implications for QoE-based optimization and differentiation strategies. The role of possible influencing factors (e.g., fatigue and boredom, test setup, content) also requires further investigation.

5. Conclusions and future work

In this paper, we have shared findings from an experimental user study on QoE in the context of adaptive video streaming. The objectives of the study presented in this paper were twofold: we first of all aimed to investigate QoE in the context of different realistic adaptive video streaming profiles. The video streams were delivered according to three different bitrate profiles, representing realistic bandwidth conditions (i.e., constant high bitrate, constant low bitrate, changing bitrate resulting in adaptive video quality). In order to also contribute to the ongoing discussions on the measurement of QoE, both traditional and more alternative measures of QoE (linking up to [5]) were included. Secondly, we wanted to explore the possible influence of specific contextual variables on QoE. More concretely, the test took place in two usage scenarios representing different social settings (i.e., alone, dedicated internet access vs. together with others, notion of sharing internet access with others).

Although our findings are valid for the study presented here and cannot simply be generalized, they point to differences between the three bitrate profiles in terms of QoE (both considering the traditional and the alternative QoE measures). We found indications that a lower, constant bitrate is preferred over the changing bitrate which is - on average - higher. The latter is associated with lower delight, higher annoyance and higher odds to be evaluated as 'not acceptable'. This finding is in line with previous studies such as [11] and confirms that a constant, fluent video playout is important and that variations in video playout are annoying. This is the case for both separate usage scenarios. However, based on our findings from this first study, we cannot claim that these differences are significant from a statistical point of view. Interestingly, the majority of the videos were evaluated as acceptable to watch, and indicate a higher impairment tolerance than when only considering the overall quality ratings. Although we found no significant influence of the social setting when considering traditional QoE evaluation measures, significant differences between both settings were detected when evaluating QoE in terms of delight and annoyance. Moreover, it could be observed that QoE was consistently better in scenario 1.

This study was only a first step and further research is needed to better understand user preferences in the context of adaptive video streaming. In future work, we aim to further zoom in on the possible influence of the social setting (through different setups, e.g., in naturalistic settings with higher ecological validity). Additionally, we aim to take a closer look at the interplay between content likeability, mood and QoE; and to investigate how traditional and alternative measures of QoE relate to each other. In our future work, we aim to also consider other adaptive streaming solutions and terminal devices, additional bitrate profiles (e.g., from low to high bitrate), alternative test setups (including 'fake streaming', with locally stored content and encoded impairments) as well as other types of content.

References

[1] Cisco Visual Networking Index: Forecast and Methodology, 2010- 2015. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

[2] Oyman O, Singh S (2012) Quality of experience for HTTP adaptive streaming services. Communications Magazine, IEEE 50 (4):20-27. doi:10.1109/mcom.2012.6178830

[3] Siglin T (2013) The State of OTT Content Delivery 2013. Streaming Media Magazine (2)

[4] Sodagar I (2011) The MPEG-DASH Standard for Multimedia Streaming Over the Internet. Multimedia, IEEE 18 (4):62-67. doi:10.1109/mmul.2011.71

[5] DASH Industry Forum, inc. (2013, May) Survey of European Broadcasters on MPEG-DASH Whitepaper http://dashif.org/w/2013/05/Survey-of-the-European-Broadcasters-on-MPEG-DASH-Whitepaper-V2.1.pdf. Accessed June 16, 2013

[6] Le Callet P, Möller S, Perkis A (eds) (2012) Qualinet White Paper on Definitions of Quality of Experience, version 1.1, June 3, 2012. European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Lausanne

[7] ITU (2008) ITU-T Rec. P.10 Vocabulary for performance and quality of service, Amendment 2: New definitions for inclusion in Recommendation ITU-T P.10/G.100. International Telecommunication Union, Geneva

[8] De Moor K (2012) Are engineers from Mars and users from Venus? Bridging gaps in Quality of Experience research: reflections on and experiences from an interdisciplinary journey. Unpublished doctoral dissertation, Ghent University: Department of Communication Studies

[9] De Moor K, Rios Quintero M, Strohmeier D, Raake A (2013) Evaluating QoE by means of traditional and alternative subjective measures: an exploratory 'living room lab' study on IPTV. Paper presented at the 4th International Workshop on Perceptual Quality of Systems (PQS 2013) Vienna, Austria

[10] Singh KD, Hadjadj-Aoul Y, Rubino G Quality of experience estimation for adaptive HTTP/TCP video streaming using H.264/AVC. In: Consumer Communications and Networking Conference (CCNC), 2012 IEEE, 14-17 Jan. 2012 2012. pp 127-131. doi:10.1109/ccnc.2012.6181070

[11] Robinson DC, Jutras Y, Craciun V (2012) Subjective Video Quality Assessment of HTTP Adaptive Streaming Technologies. Bell Labs Technical Journal 16 (4):5-23. doi:10.1002/bltj.20531

[12] Microsoft, inc. (January 2013) IIS Smooth Streaming Transport Protocol, [MS-SSTR] v20130118. http://msdn.microsoft.com/en-us/library/ff469518.aspx.

[13] Pantos M (2013, April) Apple HLS (HTTP Live Streaming). IETF draft. http://tools.ietf.org/html/draft-pantos-http-live-streaming-11.

[14] Adobe systems, inc. HTTP Dynamic Streaming http://www.adobe.com/products/httpdynamicstreaming/.

[15] ISO, inc. (2012, April) ISO MPEG-DASH Specification (ISO/IEC DIS 23009-1:2012) http://dashif.org/mpeg-dash/.

[16] Akhshabi S, Begen AC, Dovrolis C An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. In: ACM Multimedia Systems (MMSys), 2011. ACM, pp 157-168

[17] Desmet PMA, Vastenburg MH, Van Bel D, Romero NA Pick-A-Mood; development and application of a pictorial mood-reporting instrument. In: Brassett J, Hekkert P, Ludden G, Malpass M, McDonnell J (eds) 8th International Design and Emotion Conference, Central Saint Martin College of Art & Design (11-14 September 2012), London 2012.

[18] IDC Worldwide Tablet Tracker, inc. (2013, May 1) Worldwide Tablet Market Surges Ahead on Strong First Quarter Sales - press release. http://www.idc.com/getdoc.jsp?containerId=prUS24093213 Accessed June 16 2013

References

[1] C. Bettini and D. Riboni, “Profile aggregation and policy evaluation for adaptive internet services,” in *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, 2004, pp. 290–298.

[2] R. Grieco, D. Malandrino, F. Mazzoni, and D. Riboni, “Context-aware provision of advanced internet services,” in *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, 2006, pp. 4 pp.–603.

[3] ISO, *Dynamic adaptive streaming over HTTP (DASH). ISO/IEC FCD 23001-6*, International Organization for Standardization Std. ISO/IEC FCD 23001-6, 2011.

[4] A. Metzger, K. Pohl, M. Papazoglou, E. Di Nitto, A. Marconi, and D. Karastoyanova, “Research challenges on adaptive software and services in the future internet: towards an s-cube research roadmap,” in *Software Services and Systems Research - Results and Challenges (S-Cube), 2012 Workshop on European*, 2012, pp. 1–7.

[5] “Telco 2015, five telling years, four future scenarios,” IBM, Tech. Rep., 2010.

[6] “Cisco visual networking index: Forecast and methodology, 2010-2015,” Cisco Systems, Tech. Rep., June 2011. [Online]. Available: http://www.cisco.com/en/-US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

[7] O. Oyman and S. Singh, “Quality of experience for http adaptive streaming services,” *Communications Magazine, IEEE*, vol. 50, no. 4, pp. 20–27, 2012.

[8] *Recommendation P.10 (1998) - Vocabulary of terms on telephone transmission quality and telephone sets*, ITU-T Std.

[9] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine, “A Framework for Integrated Services Operation over Diffserv Networks,” RFC 2998 (Informational), Internet Engineering Task Force, Nov. 2000. [Online]. Available: http://-www.ietf.org/rfc/rfc2998.txt

[10] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An Architecture for Differentiated Service,” RFC 2475 (Informational), Internet Engineering Task Force, Dec. 1998, updated by RFC 3260. [Online]. Available: http://www.ietf.org/rfc/rfc2475.txt

[11] *Recommendation P.10/G.100 (2006)/Amd.1 (01/2007) - Vocabulary for performance and quality of service*, ITU-T Std.

[12] *Recommendation E.800 (09/08) - Definitions of terms related to quality of service*, ITU-T Std.

[13] *Recommendation Y.1540 (03/11) - Internet protocol data communication service - IP packet transfer and availability performance parameters*, ITU-T Std.

[14] *Recommendation Y.1541 (12/11) - Network performance objectives for IP-based services*, ITU-T Std.

[15] K. De Moor, “Are engineers from mars and users from venus?: bridging gaps in quality of experience research: reflections on and experiences from an interdisciplinary journey,” Ph.D. dissertation, Ghent University, 2012.

[16] P. Le Callet, S. Möller, and A. Perkis, “Qualinet white paper on definitions of quality of experience,” June 2012.

[17] E. M. Rogers, *Diffusion of innovations*, 4th ed. Free Press, New York, 1962.

[18] Akamai technologies, company profile online at http://www.akamai.com.

[19] J. Brutlag. (2009) Speed matters for google web search. [Online]. Available: http://code.google.com/speed/files/delayexp.pdf

[20] R. K. Sitaraman, “Network performance: Does it really matter to users and by how much?” in *Proceedings of the 5th International Conference on Communication Systems and Networks (COMSNETS 2013)*, 2013, pp. 1–10.

[21] D. Xu, S. S. Kulkarni, C. Rosenberg, and H. keung Chai, “A cdn-p2p hybrid architecture for cost-effective streaming media distribution,” *Computer Networks*, vol. 44, pp. 353–382, 2004.

[22] C. Huang, A. Wang, J. Li, and K. W. Ross, “Understanding hybrid cdn-p2p: why limelight needs its own red swoosh,” pp. 75–80, 2008.

[23] G. Fortino and W. Russo, “Using p2p, {GRID} and agent technologies for the development of content distribution networks,” *Future Generation Computer Systems*, vol. 24, no. 3, pp. 180 – 190, 2008. [Online]. Available: http://-www.sciencedirect.com/science/article/pii/S0167739X07001021

[24] MUSE. (2004-2008) Multi service access everywhere. [Online]. Available: http://www.ist-muse.org/

[25] ALPHA. (2008-2010) Architectures for flexible photonic home and access networks. [Online]. Available: http://www.ict-alpha.eu/

[26] OMEGA. (2008-2010) The home gigabit access project. [Online]. Available: http://www.ict-omega.eu/

[27] H. Dequeker, T. V. Caenegem, K. Struyve, E. Gilon, B. D. Vleeschauwer, P. Simoens, and W. V. de Meerssche, “Advanced features for mm enabled access platform (muse project deliverable),” Alcatel Bell and IBBT - IBCN, Tech. Rep., December 2006. [Online]. Available: http://www.ist-muse.org/-deliverables_list.htm

[28] B. D. Vleeschauwer, W. V. de Meerssche, P. Simoens, F. D. Turck, B. Dhoedta, and P. Demeester, “Enabling autonomic access network qoe management through tcp monitoring,” in *Proceedings of the First IEEE Workshop on Autonomic Communications and Management (ACNM)*, 2007.

[29] P. Simoens, B. D. Vleeschauwer, W. V. de Meerssche, F. D. Turck, B. Dhoedta, and P. Demeester, “Rtp connection monitoring for enabling autonomic access network qos management,” in *Proceedings of the 12th European Conference on Networks and Optical Communications (NOC)*, 2007.

[30] B. D. Vleeschauwer, P. Simoens, W. V. de Meerssche, S. Latré, F. D. Turck, and S. V. den Bergheb, “Autonomic qoe optimization on the access node,” in *Proceedings of the Broadband Europe*, 2007.

[31] S. Latr´e, P. Simoens, B. De Vleeschauwer, and V. de Meerssche, “An autonomic architecture for optimizing qoe in multimedia access networks,” *Computer Networks 53*, vol. 10, p. 1587–1602, 2009.

[32] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *Multimedia, IEEE*, vol. 18, no. 4, pp. 62 –67, april 2011.

[33] Y. Royon and S. Frénot, "Multiservice Home Gateways: Business Model, Execution Environment, Management Infrastructure," *IEEE Communications Magazine*, vol. 45, no. 10, pp. 122–128, 2007.

[34] OSGi. Open software gateway initiative. [Online]. Available: http://-www.osgi.org/

[35] Universal plug and play forum. UPnP Forum. [Online]. Available: http://-www.upnp.org

[36] Upnp qos architecture:3. UPnP Forum. [Online]. Available: http://upnp.org/-specs/qos/qos3/

[37] V. Suraci, G. Oddi, N. Mattiacci, and A. Angelucci, "Admission control and drop strategies in a upnp-qos controlled home network." in *PIMRC 2010*. IEEE, 2010, pp. 2811–2816. [Online]. Available: http://dblp.uni-trier.de/db/conf/-pimrc/pimrc2010.html#SuraciOMA10

[38] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633 (Informational), Internet Engineering Task Force, Jun. 1994. [Online]. Available: http://www.ietf.org/rfc/rfc1633.txt

[39] G. Ash and D. McDysan, "Generic connection admission control (gcac) algorithm specification for ip/mpls networks," Internet Engineering Task Force, 2012, updated by RFCs 1349, 4379, 5884, 6093.

[40] P. Eardley, "Pre-Congestion Notification (PCN) Architecture," RFC 5559 (Informational), Internet Engineering Task Force, Jun. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5559.txt

[41] S. Latre, K. Roobroeck, T. Wauters, and F. Turck, "Protecting video service quality in multimedia access networks through pcn," *Communications Magazine, IEEE*, vol. 49, no. 12, pp. 94–101, 2011.

[42] J.-B. Lee and H. Kalva. Video coding techniques and standards. Springer. [Online]. Available: http://www.springerreference.com/docs/html/chapterdbid/-73083.html

[43] *MPEG-1 / ISO/IEC 11172-2*, Moving Pictures Expert Group (MPEG), ISO/IEC Working Group Std.

[44] *H-261 (03/93) Video codec for audiovisual services at p x 64 kbit/s*, ITU-T VCEG Std.

[45] *MPEG-2 / ISO/IEC 11172-2*, Moving Pictures Expert Group (MPEG), ISO/IEC Working Group Std.

[46] *H-262 (02/12) Information technology - Generic coding of moving pictures and associated audio information: Video*, ITU-T VCEG Std.

[47] *MPEG-4 / ISO/IEC 14496-2*, Moving Pictures Expert Group (MPEG), ISO/IEC Working Group Std.

[48] *H-263 (01/05) Video coding for low bit rate communication*, ITU-T VCEG Std.

[49] *MPEG-4 AVC / ISO/IEC 14496-15*, Moving Pictures Expert Group (MPEG), ISO/IEC Working Group Std.

[50] *H-264 (04/13) Advanced video coding for generic audiovisual services*, ITU-T VCEG Std.

[51] *MPEGH / ISO/IEC 23008*, Moving Pictures Expert Group (MPEG), ISO/IEC Working Group Std.

[52] *H.265 (04/13) High efficiency video coding*, ITU-T VCEG Std.

[53] G. Bulycs, F. Aghdasi, and M. Bourges-Sevenier. (2005) Understanding mpeg-4: Technologies, advantages, and markets. MPEG Industry Forum.

[54] V. Jacobson, "Congestion avoidance and control," *SIGCOMM Comput. Commun. Rev.*, vol. 18, pp. 314–329, August 1988. [Online]. Available: http://-doi.acm.org/10.1145/52325.52356

[55] ANSI, *ANSI T1.413-1998: Network and Customer Installation Interfaces — Asymmetric Digital Subscriber Line (ADSL) Metallic Interface*, ANSI Std., 1998.

[56] OECD. (2012, Dec) Fixed and wireless broadband subscriptions by technology. Online. [Online]. Available: http://www.oecd.org/sti/broadband/-oecdbroadbandportal.htm

[57] ITU-T, *Recommendation G.992.5 (05/03) - Asymmetric Digital Subscriber Line (ADSL) transceivers - Extended bandwidth ADSL2 (ADSL2plus*, ITU-T Std.

[58] ITU-T, *Recommendation G.711 - Pulse code modulation (PCM) of voice frequencies*, ITU-T Std.

[59] ETSI, *ETSI ETS 300 581-2 : Digital cellular telecommunications system (Phase 2+) (GSM); Half rate speech; Half rate speech transcoding*, ETSI Std., 4 2000.

[60] ETSI, *ETSI EN 300 961 V8.1.1 - Digital cellular telecommunications system (Phase 2+) (GSM); Full rate speech; Transcoding*, ETSI Std., 7 2000.

[61] *Recommendation G.722.2 (07/03) - Wideband coding of speech at around 16 kbit/s using Adaptive Multi-Rate Wideband (AMR-WB)*, ITU-T Std.

[62] 3GPP, *3GPP TS 26.171 - Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; General description*, 3GPP Std.

[63] N. Costa and M. Nunes, "Adaptive quality of service in voice over ip communications," in *Networking and Services, 2009. ICNS '09. Fifth International Conference on*, 2009, pp. 19–24.

[64] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889 (Proposed Standard), Internet Engineering Task Force, 1996, obsoleted by RFC 3550. [Online]. Available: http://www.ietf.org/rfc/rfc1889.txt

[65] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (rtsp)," RFC 2326 (Proposed Standard), Internet Engineering Task Force, Apr. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2326.txt

[66] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Rtp: A transport protocol for real-time applications," RFC 3550 (Standard), Internet Engineering Task Force, 2003, updated by RFCs 5506, 5761, 6051, 6222. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[67] *ISO/IEC 23009-1:2012(E): Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats*, ISO/IEC Std., 4 2012.

[68] (2013, May) Adaptive streaming for you and youtube. Google I/0. [Online]. Available: https://developers.google.com/events/io/sessions/328181796

[69] A. Rao, A. Legout, Y.-s. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*,

ser. CoNEXT ’11. New York, NY, USA: ACM, 2011, pp. 25:1–25:12. [Online]. Available: http://doi.acm.org/10.1145/2079296.2079321

[70] H. Riiser, H. S. Bergsaker, P. Vigmostad, P. Halvorsen, and C. Griwodz, “A comparison of quality scheduling in commercial adaptive http streaming solutions on a 3g network,” in *Proceedings of the 4th Workshop on Mobile Video*, ser. MoVid ’12. New York, NY, USA: ACM, 2012, pp. 25–30. [Online]. Available: http://doi.acm.org/10.1145/2151677.2151684

[71] J. Strauss, D. Katabi, and F. Kaashoek, “A measurement study of available bandwidth estimation tools,” in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, ser. IMC ’03. New York, NY, USA: ACM, 2003, pp. 39–44. [Online]. Available: http://doi.acm.org/10.1145/-948205.948211

[72] S. Ekelin, M. Nilsson, E. Hartikainen, A. Johnsson, J.-E. Mangs, B. Melander, and M. Bjorkman, “Real-time measurement of end-to-end available bandwidth using kalman filtering,” in *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, 2006, pp. 73–84.

[73] C. Liu, I. Bouazizi, and M. Gabbouj, “Rate adaptation for adaptive http streaming,” in *the second annual ACM conferenceProceedings of the second annual ACM conference on Multimedia systems - MMSys 11*, ACM Press. San Jose, CA, USANew York, New York, USA: ACM Press, 2011.

[74] L. De Cicco, S. Mascolo, and V. Palmisano, “Feedback control for adaptive live video streaming,” in *Proceedings of the second annual ACM conference on Multimedia systems*, ser. MMSys ’11. New York, NY, USA: ACM, 2011, pp. 145–156. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943573

[75] C. Krasic, K. Li, and J. Walpole, “The case for streaming multimedia with tcp,” in *In 8th International Workshop on Interactive Distributed Multimedia Systems (iDMS 2001*, 2001, pp. 213–218.

[76] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, “Multimedia streaming via tcp: An analytic performance study,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 4, no. 2, pp. 16:1–16:22, May 2008. [Online]. Available: http://-doi.acm.org/10.1145/1352012.1352020

[77] W. Stevens, “Tcp slow start, congestion avoidance, fast retransmit, and fast recovery algorithms,” RFC 2001 (Proposed Standard), Internet Engineering Task Force, Jan. 1997, obsoleted by RFC 2581. [Online]. Available: http://-www.ietf.org/rfc/rfc2001.txt

[78] K. S. Habibullah Jamal, “Performance analysis of tcp congestion control algorithms,” I. J. O. COMPUTERS and COMMUNICATIONS, Eds., vol. Volume 2, no. Issue 1, 2008.

[79] K. Ramakrishnan, S. Floyd, and D. Black, “The addition of explicit congestion notification (ecn) to ip,” RFC 3168 (Proposed Standard), Internet Engineering Task Force, Sep. 2001, updated by RFCs 4301, 6040. [Online]. Available: http://www.ietf.org/rfc/rfc3168.txt

[80] D. Katabi, M. Handley, and C. Rohrs, “Congestion control for high bandwidth-delay product networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 89–102, Aug. 2002. [Online]. Available: http://doi.acm.org/10.1145/-964725.633035

[81] L. Stewart, D. A. Hayes, G. Armitage, M. Welzl, and A. Petlund, “Multimedia-unfriendly tcp congestion control and home gateway queue management,” in *Proceedings of the second annual ACM conference on Multimedia systems*, ser. MMSys ’11. New York, NY, USA: ACM, 2011, pp. 35–44. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943558

[82] S. Ha, I. Rhee, and L. Xu, “Cubic: a new tcp-friendly high-speed tcp variant,” *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, pp. 64–74, Jul. 2008. [Online]. Available: http://doi.acm.org/10.1145/1400097.1400105

[83] K. Kaneko, T. Fujikawa, Z. Su, and J. Katto, “Tcp-fusion: A hybrid congestion control algorithm for high-speed networks,” Proceedings of PFLDnet, pp.31-36, Feb 2007.

[84] K. Tan, J. Song, Q. Zhang, and M. Sridharan, “A compound tcp approach for high-speed and long distance networks,” in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, 2006, pp. 1–12.

[85] D. Cavendish, G. Wataabe, K. Kumazoe, D. Nobayashi, T. Ikenaga, and Y. Oie, “Performance characterization of streaming video over tcp variants,” in *Proceedings of INTERNET 2013 : The Fifth International Conference on Evolving Internet*, 2013, pp. 16–21.

[86] A. Blanc, K. Avrachenkov, D. Collange, and G. Neglia, “Compound tcp with random losses,” in *NETWORKING 2009*, ser. Lecture Notes in Computer Science, L. Fratta, H. Schulzrinne, Y. Takahashi, and O. Spaniol, Eds. Springer Berlin Heidelberg, 2009, vol. 5550, pp. 482–494. [Online]. Available: http://-dx.doi.org/10.1007/978-3-642-01399-7_38

[87] D. Wei, C. Jin, S. Low, and S. Hegde, “Fast tcp: Motivation, architecture, algorithms, performance,” *Networking, IEEE/ACM Transactions on*, vol. 14, no. 6, pp. 1246–1259, 2006.

[88] A. Zambelli, “iis smooth streaming technical overview,” http://www.microsoft.com/silverlight/whitepapers/, Tech. Rep., March 2009.

[89] R. Jain, *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*, ser. Wiley professional computing. Wiley, 1991. [Online]. Available: http://-books.google.no/books?id=CN1QAAAAMAAJ

[90] P. J. Emstad, P. E. Heegaard, B. E. Helvik, and L. Paquereau, *Dependability and performance in information and communication systems. Fundamentals*. Akademika Forlag, 2013.

[91] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[92] G. Riley and T. Henderson, “The ns-3 network simulator,” in *Modeling and Tools for Network Simulation*, K. Wehrle, M. GÃ¼neÅŸ, and J. Gross, Eds. Springer Berlin Heidelberg, 2010, pp. 15–34. [Online]. Available: http://-dx.doi.org/10.1007/978-3-642-12331-3_2

[93] G. Birtwistle, *Demos - A system for Discrete Event Modelling on Simula*. School of Computer Science, University of Sheffeld,, July 1997. [Online]. Available: http://staffwww.dcs.shef.ac.uk/people/G.Birtwistle/research/demos.pdf

[94] R. Morris, E. Kohler, J. Jannotti, and M. F. Kaashoek, “The click modular router,” *Operating Systems Review 34*, vol. 5, p. 217–231, December 1999.

[95] M. Pantos. “http live streaming”. ietf draft november 2010. visited: 16.02.2011 <http://tools.ietf.org/html/draft-pantos-http-live-streaming-05>.

[96] R. H. Inc. Cygwin. [Online]. Available: http://www.redhat.com/services/-custom/cygwin/

[97] A. Botta, A. Dainotti, and A. Pescape, “Do you trust your software-based traffic generator?” *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 158–165, 2010.

[98] R. Prasad, M. Jain, and C. Dovrolis, “Effects of interrupt coalescence on network measurements,” in *In Proceedings of PAM ’04*, 2004.

[99] J. Laine, S. Saaristo, and R. Prior, “Real-time udp data emitter (rude) and collector for rude (crude),” http://rude.sourceforge.net/, Jan 2000. [Online]. Available: http://rude.sourceforge.net/

[100] S. Ubik, V. Smotlach, S. Saaristo, and J. Laine, “Low-cost precise qos measurement tool, cesnet technical report number 7/2001,” CESNET, Tech. Rep., 2001. [Online]. Available: http://www.cesnet.cz/doc/techzpravy/2001/07/-qosplot.pdf

[101] B. J. Villa and P. E. Heegaard, “Monitoring and control of qoe in media streams using the click software router,” in *NIK2010, Norway. ISBN 978-82-519-2702-4.*, vol. 1, November 2010, pp. 24–33. [Online]. Available: http://tapironline.no/-last-ned/332

[102] B. J. Villa and P. E. Heegaard, “Towards knowledge-driven qoe optimization in home gateways,” May 2011.

[103] K. Nichols, V. Jacobson, and L. Zhang, *A Two-bit Differentiated Services Architecture for the Internet*, RFC 2638 (Informational), Internet Engineering Task Force Std. 2638, Jul. 1999. [Online]. Available: http://www.ietf.org/rfc/-rfc2638.txt

[104] B. J. Villa and P. E. Heegaard, “A monitor plane component for adaptive video streaming,” in *NIK2011, Norway. ISSN 1892-0713*, vol. 1, November 2011, pp. 145–154. [Online]. Available: http://www.tapironline.no/fil/vis/857

[105] K. Haugene and A. Jacobsen, “Network based qoe optimization for “over the top” services,” Master’s thesis, NTNU, 2011.

[106] B. J. Villa, K. D. Moor, P. E. Heegaard, and A. Instefjord, “Investigating quality of experience in the context of adaptive video streaming: findings from an experimental user study,” in *Norsk Informatikkonferanse NIK 2013, Norway.*, ser. ISSN 1892-0713, vol. 1, November 2013, pp. 122–133.

[107] B. J. Villa and P. E. Heegaard, “Improving perceived fairness and qoe for adaptive video streams,” *ICNS 2012*, March 2012.

[108] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, “Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification,” RFC 2205 (Proposed Standard), Internet Engineering Task Force, Sep. 1997, updated by RFCs 2750, 3936, 4495, 5946. [Online]. Available: http://www.ietf.org/rfc/-rfc2205.txt

[109] M. Z. Chowdhury, S. Choi, Y. M. Jang, K.-S. Park, and G. I. Yoo, “Dynamic SLA negotiation using bandwidth broker for femtocell networks,” in *Ubiquitous and Future Networks, 2009. ICUFN 2009. First International Conference on*. IEEE, Jun. 2009, pp. 12–15. [Online]. Available: http://dx.doi.org/10.1109/-icufn.2009.5174276

[110] M. Z. Chowdhury, Y. M. Jang, and Z. J. Haas, “Network evolution and qos provisioning for integrated femtocell/macrocell networks,” *CoRR*, vol. abs/1009.2368, 2010. [Online]. Available: http://dblp.uni-trier.de/db/journals/-corr/corr1009.html#abs-1009-2368

[111] K. I. Kim, S. H. Kim, J. M. Moon, and Y. J. Kim, “Hierarchical admission control scheme for supporting mobility in mobile ip,” in *Proceedings of the IEEE MILCOM 2002*, 2002, pp. 431–435.

[112] “Hgi-rd027-r3 9, home gateway qos module requirements,” HGi - Home Gateway Initiative, Tech. Rep., Dec 2012.

[113] “Hgi-gd006-r2, hgi guideline paper: Ims enabled hg,” HGi - Home Gateway Initiative, Tech. Rep., May 2009.

[114] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, “B4: experience with a globally-deployed software defined wan,” in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, ser. SIGCOMM ’13. New York, NY, USA: ACM, 2013, pp. 3–14. [Online]. Available: http://dx.doi.org/-10.1145/2486001.2486019

[115] J. Rajahalme, A. Conta, B. Carpenter, and S. Deering, “IPv6 Flow Label Specification,” RFC 3697 (Proposed Standard), Internet Engineering Task Force, Mar. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3697.txt

[116] L. Chenghao, I. Bouazizi, and M. Gabbouj, “Segment duration for rate adaptation of adaptive http streaming,” in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, july 2011, pp. 1 –4.

[117] C. Liu, I. Bouazizi, and M. Gabbouj, “Parallel adaptive http media streaming,” in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, 31 2011-aug. 4 2011, pp. 1 –6.

[118] B. J. Villa and P. E. Heegaard, “Detecting period and burst durations in video streaming by means of active probing,” in *Proceedings of the 3rd International Conference on Computer Communication and Management (ICCCM 2013)*, 2013.

[119] J. Postel, “Transmission control protocol,” RFC 793 (Standard), Internet Engineering Task Force, Sep. 1981, updated by RFCs 1122, 3168, 6093. [Online]. Available: http://www.ietf.org/rfc/rfc793.txt

[120] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, “A model for tcp-based video streaming.”

[121] W. K. Lai and C. Y. Pan, “Achieving inter-session fairness for layered video multicast,” in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 3, 2001, pp. 935–939 vol.3.

[122] Y. H. Jung and Y. Choe, “Resource-aware and quality-fair video-streaming using multiple adaptive {TCP} connections,” *Computers & Electrical Engineering*, vol. 36, no. 4, pp. 702 – 717, 2010, <ce:title>Signal Processing and Communication Systems</ce:title>. [Online]. Available: http://-www.sciencedirect.com/science/article/pii/S0045790608001201

[123] V. Joseph and G. de Veciana, “Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs.” in *INFOCOM*, A. G. Greenberg and K. Sohraby, Eds. IEEE, 2012, pp. 567–575.

[Online]. Available: http://dblp.uni-trier.de/db/conf/infocom/-infocom2012.html#JosephV12

[124] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, “Towards network-wide QoE fairness using openflow-assisted adaptive video streaming,” in *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*, ser. FhMN ’13. New York, NY, USA: ACM, 2013, pp. 15–20. [Online]. Available: http://dx.doi.org/10.1145/2491172.2491181

[125] J. Jiang, V. Sekar, and H. Zhang, “Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive,” in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, ser. CoNEXT ’12. New York, NY, USA: ACM, 2012, pp. 97–108. [Online]. Available: http://doi.acm.org/10.1145/2413176.2413189

[126] B. J. Villa, P. E. Heegaard, and A. Instefjord, “Improving fairness for adaptive http video streaming,” in *EUNICE*, 2012, pp. 183–193.

[127] M. Marcon, M. Dischinger, K. Gummadi, and A. Vahdat, “The local and global effects of traffic shaping in the internet,” in *Communication Systems and Networks (COMSNETS), 2011 Third International Conference on*, 2011, pp. 1–10.

[128] M. F. Alam, M. Atiquzzaman, and M. A. Karim, “Traffic shaping for mpeg video transmission over the next generation internet,” *Comput. Commun.*, vol. 23, no. 14-15, pp. 1336–1348, Aug. 2000. [Online]. Available: http://-dx.doi.org/10.1016/S0140-3664(00)00180-8

[129] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, and J. Lopez-Soler, “Analysis and modelling of youtube traffic,” *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 4, pp. 360–377, 2012. [Online]. Available: http://dx.doi.org/10.1002/ett.2546

[130] M. Ghobadi, Y. Cheng, A. Jain, and M. Mathis, “Trickle: rate limiting youtube video streaming,” in *Proceedings of the 2012 USENIX conference on Annual Technical Conference*, ser. USENIX ATC’12. Berkeley, CA, USA: USENIX Association, 2012, pp. 17–17. [Online]. Available: http://dl.acm.org/-citation.cfm?id=2342821.2342838

[131] S. Singh, J. G. Andrews, and G. de Veciana, “Interference shaping for improved quality of experience for real-time video streaming.” *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 7, pp. 1259–1269, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/jsac/jsac30.html#SinghAV12

[132] *Transparent end-to-end packet switched streaming service (PSS); Progressive download and dynamic adaptive streaming over HTTP (3GP-DASH), Release 10, TS 26.247 v 10.1.0*, 3GPP Std., June 2011.

[133] J. Esteban, S. A. Benno, A. Beck, Y. Guo, V. Hilt, and I. Rimac, “Interactions between http adaptive streaming and tcp,” in *Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video*, ser. NOSSDAV ’12. New York, NY, USA: ACM, 2012, pp. 21–26. [Online]. Available: http://doi.acm.org/10.1145/2229087.2229094

[134] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. C. Begen, “Server-based traffic shaping for stabilizing oscillating adaptive streaming players,” in *Proceeding of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, ser. NOSSDAV ’13. New York, NY,

USA: ACM, 2013, pp. 19–24. [Online]. Available: http://doi.acm.org/10.1145/-2460782.2460786

[135] R. Houdaille and S. Gouache, “Shaping http adaptive streams for a better user experience,” in *MMSys*, 2012, pp. 1–9.

[136] T. Bu, N. Duffield, F. L. Presti, and D. Towsley, “Network tomography on general topologies,” *SIGMETRICS Perform. Eval. Rev.*, vol. 30, no. 1, pp. 21–30, Jun. 2002. [Online]. Available: http://doi.acm.org/10.1145/511399.511338

[137] A. Pasztor and D. Veitch, “Active probing using packet quartets,” in *In ACM SIGCOMM Internet Measurement Workshop*, 2002, pp. 293–305.

[138] M. Jain and C. Dovrolis, “Ten fallacies and pitfalls on end-to-end available bandwidth estimation,” in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ser. IMC ’04. New York, NY, USA: ACM, 2004, pp. 272–277. [Online]. Available: http://dx.doi.org/10.1145/1028788.1028825

[139] K. C. Claffy, G. C. Polyzos, and H.-W. Braun, “Application of sampling methodologies to network traffic characterization,” in *Conference proceedings on Communications architectures, protocols and applications*, ser. SIGCOMM ’93. New York, NY, USA: ACM, 1993, pp. 194–203. [Online]. Available: http://doi.acm.org/10.1145/166237.166256

[140] J. Liu, D. Zhang, and J. Jin, “On modeling and accuracy analysis of the available bandwidth measurement based-on packet-pair sampling.” *IJCNS*, vol. 1, no. 2, pp. 168–176, 2008. [Online]. Available: http://dblp.uni-trier.de/db/journals/-ijcns/ijcns1.html#LiuZJ08

[141] O. Goga and R. Teixeira, “Speed measurements of residential internet access,” in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, N. Taft and F. Ricciato, Eds. Springer Berlin Heidelberg, 2012, vol. 7192, pp. 168–178. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-28537-0_17

[142] L. Angrisani, S. D’Antonio, M. Esposito, and M. Vadursi, “Techniques for available bandwidth measurement in ip networks: a performance comparison,” *Comput. Netw.*, vol. 50, no. 3, pp. 332–349, Feb. 2006. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2005.06.003

[143] W. Ma, J. Yan, and C. Huang, “Adaptive sampling methods for network performance measurement under voice traffic,” in *Proc. of IEEE ICC*, 2004, pp. 1129–1134.

[144] R. W. Wolff, “Poisson arrivals see time averages,” *Operations Research*, vol. 30, pp. 223–231, 1982.

[145] F. Baccelli, S. Machiraju, D. Veitch, and J. C. Bolot, “The role of pasta in network measurement,” *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 231–242, Aug. 2006. [Online]. Available: http://doi.acm.org/10.1145/-1151659.1159940

[146] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, “Tussle in cyberspace: defining tomorrow’s internet,” *IEEE/ACM Trans. Netw.*, vol. 13, no. 3, pp. 462–475, Jun. 2005. [Online]. Available: http://dx.doi.org/10.1109/-TNET.2005.850224

[147] J. Day, I. Matta, and K. Mattar, “Networking is ipc: A guiding principle to a better internet,” in *Proceedings of the 2008 ACM CoNEXT Conference*, ser. CoNEXT ’08. New York, NY, USA: ACM, 2008, pp. 67:1–67:6. [Online]. Available: http://doi.acm.org/10.1145/1544012.1544079