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

Bjørn J. Villa[1] and Poul E. Heegaard[2]

Department of Telematics,
Norwegian Institute of Science and Technology
{bjorn.villa, poul.heegaard}@item.ntnu.no

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.

This paper was presented at the NIK 2010 conference. For more information see http://www.nik.no/

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.

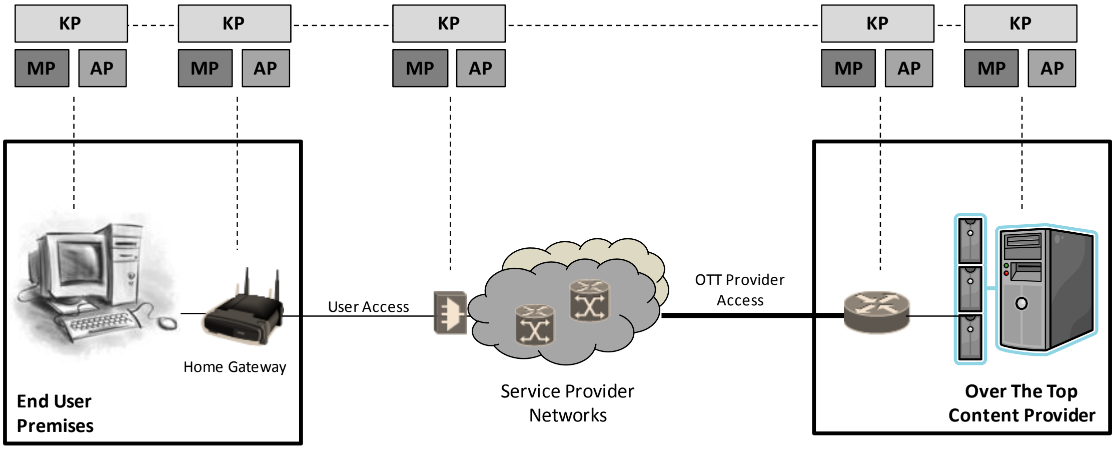

Figure 1 Reference model for QoE optimization research

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.

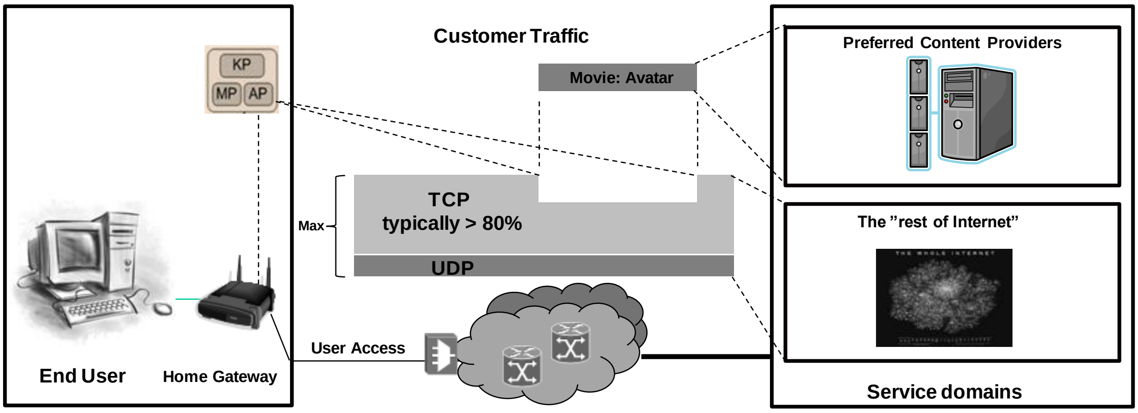

Figure 2 TCP Rate Control as AP component in Home Gateway

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.

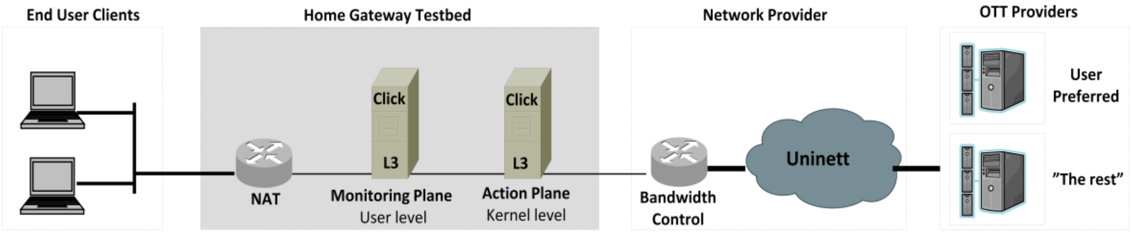

Figure 3 Home Gateway Testbed

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.

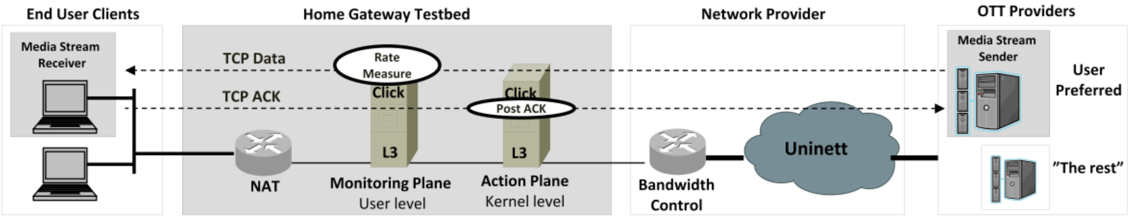

Figure 4 Location of PostACK and Rate measurement functions

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.

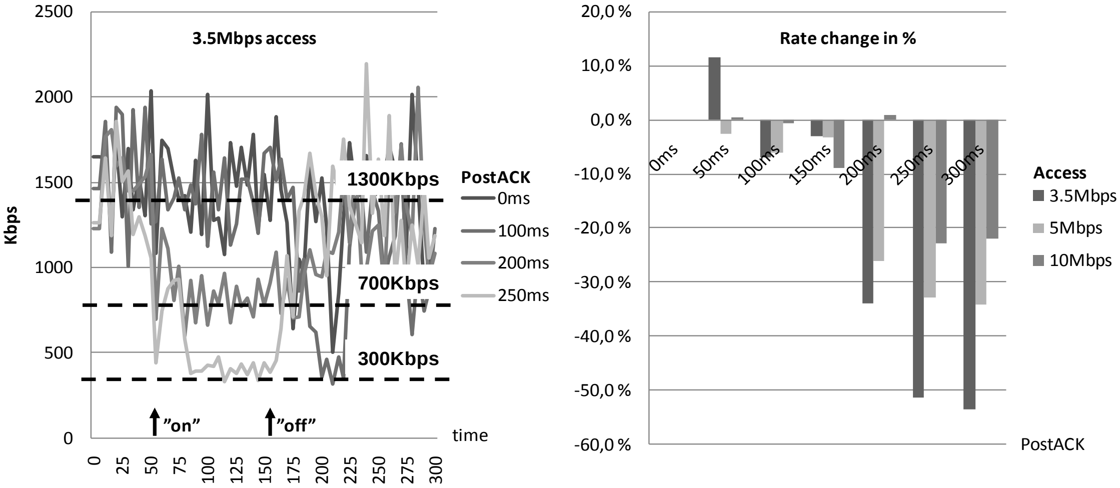

Figure 5 PostACK effect on Adaptive HTTP Streaming

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

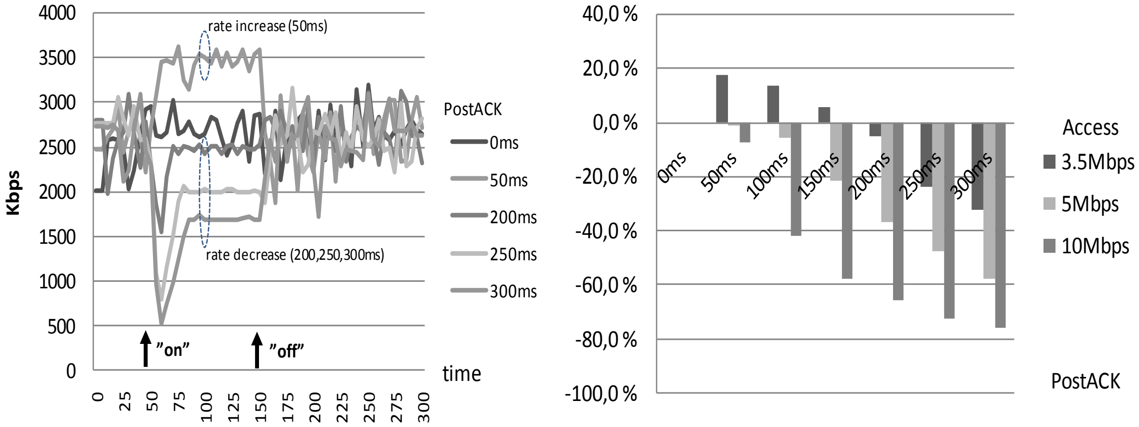

Figure 6 PostACK effect on FTP download

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 $RWIN_{bits}/RTT_2$ 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.