On the Enhancement of QoE for IPTV Services through Knowledge Plane Deployment

Bart De Vleeschauwer, Wim Van de Meerssche, Pieter Simoens, Filip De Turck, Bart Dhoedt, Piet Demeester
Ghent University - IBBT - IMEC, Department of Information Technology
Gaston Crommenlaan 8 bus 201, 9050 Gent, Belgium
Bart.DeVleeschauwer@intec.ugent.be

Edith Gilon, Kris Struyve, Tom Van Caenegem
Alcatel Research & Innovation
Copernicuslaan 50, B-2018 Antwerpen, Belgium

Abstract

To enhance the QoE (Quality of Experience) of multimedia services like IPTV (Internet Protocol TeleVision), video on demand, high speed Internet, gaming and VoIP (Voice over IP), we research the application of a monitor plane and a knowledge plane to the access network. The former is responsible for gathering monitor information along the end-to-end path between edge router and end-device, the latter for determining the root cause of problems and suggesting a solution to anomalous behavior. The logical architecture of these two planes is outlined and a number of techniques to monitor the QoE of a service at the access node are discussed. Finally, we present some evaluation results for the scalability of connection tracking that indicate the applicability of the proposed techniques.

Introduction

As broadband technology is penetrating the homes of tens of millions of users, one can but conclude that the driving factors of this broadband deployment are the services that can be delivered on top of this technology. The services that are being offered today go far beyond the Best Effort services of a decade ago. The Internet technologies are being used to form an infrastructure that delivers data, audio and video traffic at the same time. Essential for all these services is their Quality of Experience (QoE), a general term describing the quality of the services as it is perceived by the user. Multimedia services like video on demand and IPTV are particularly sensitive to packet loss, delay and jitter and often require a substantial amount of bandwidth, so a mechanism must be in place to detect QoE degradation and to react appropriately to restore the QoE.

In [1], the concept of a knowledge plane in the Internet was introduced, as an enhanced network that enables the automatic detection and recovery of faults. Here, we apply this paradigm to the access network and propose to use a two layer approach to achieve the expected behavior. The first layer is a monitor plane which gathers monitor information on the QoE of the services along the whole path from edge router until the actual end device. On top of this monitor plane, a knowledge plane will be deployed that is able to identify problems and to locate the actual cause of the QoE decrease. Additionally, the knowledge plane can propose a solution to restore the QoE of the service.

The research into a monitoring plane and knowledge plane for the access network is part of a broader and more comprehensive vision of future service-aware access networks that is under study in the integrated research project MUSE [2]. The overall goal of the MUSE project is the research and development of a future, low cost, multi-service broadband access network. MUSE is studying the QoE requirements and architecture for various services. The work reported here focuses on the architecture and several existing monitoring techniques that support the development of the user experience for a number of services. More specifically, this paper focuses on how to monitor services and devices between the access node and the end-device. In many cases, the cause of QoE degradation is situated in this part.

This paper is outlined as follows: we start with an overview of the topology of the access network and present the architecture and interaction between monitor plane and knowledge plane. A number of services are highlighted as use-cases for the knowledge plane. We then discuss a number of techniques to monitor the impact on the QoE by the connection between access node and end-device. More specifically, we discuss how active and passive monitoring can be used to infer information on the quality of the connections. Finally, we present some results on the overhead of TCP connection tracking.

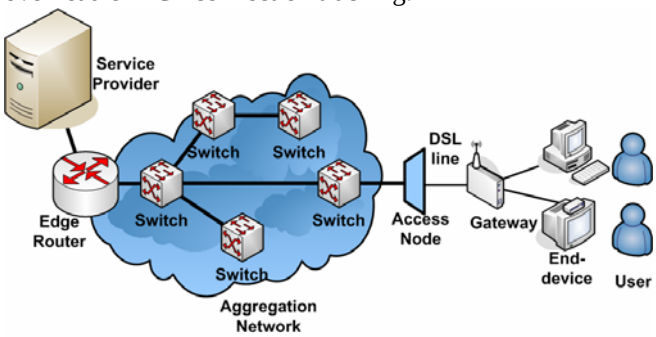

Fig. 1: Connection between service provider and user in the access network

Access Network Architecture

When observing the connection between the service provider and the device that is actually presenting the content to the user, there are a number of devices with a distinct functionality. We divide these into the following categories:

- Application servers: The service provider can have a number of servers connected to the edge router in the access network. Some examples are a video-on-demand server or a server for broadcast TV.

ISBN10 : 907654607X
ISBN13 : 9789076546070

- Edge router: The edge router provides the connectivity to the public Internet and to other services that are accessed from the end-devices.
- Aggregation network: To connect the edge router to the access nodes of the subscribers, an aggregation network is used. Traffic in this network is mostly prioritized in different classes.
- Access node: The access node terminates the aggregation network and serves as an access multiplexer to provide the connectivity with the clients. We assume that the access node is also equipped with the ability to intercept and parse traversing packets and can decide to alter/drop these, based on dedicated L3 filtering and processing capabilities.
- User connection: The user connection we consider in this paper is DSL based.
- Home gateway: The home gateway connects the access network and the home network. In addition to this, home gateways typically perform a number of additional tasks like firewalling and Network Address Translation (NAT).
- End device: The end device is located in the home network and is used directly by the user of the service; it can be a Set Top Box (STB), a game console or a standard desktop pc. The connection between the home gateway and the end device might be provided over a wireless network or via a fixed cable (e.g. ethernet).

In fig. 1 the different parts of the access network are shown.

Monitor Plane & Knowledge Plane Architecture

There are two logical layers to achieve the self-organizing and QoE optimizing access network we envision: the monitor plane (MP) and the knowledge plane (KP). In fig. 2, these two layers, their interactions and their position in the access network are shown.

Monitor Plane

The monitor plane is a logical layer that comprises all the monitor tools that are available to the access network provider. Its main function is providing information on the status of the different devices in the access network to the knowledge plane and exporting an interface so that the knowledge plane can access all this information transparently. Also, this plane realizes an efficient reduction of the huge amount of available monitoring data.

As network monitoring is a domain that has already been researched thoroughly, there is a vast amount of tools available to the monitor plane for deployment along the path from edge router to the end-device. The Simple Network Management Protocol (SNMP) [3] can be used for monitoring the aggregation network switches. This protocol is designed for communication between a central management/monitor platform and a managed device. It is possible to read and write parameters on these devices and to set traps for asynchronous communication. The structure of the parameters that can be set is specified in a Management Information Base (MIB), like the MIBs defined by the rmonmib working group of the IETF [4]. IPFIX [5] is another technique to monitor the switches and the edge routers. It has also been specified in the IETF. It allows tracking the behavior of individual flows, which are identified by the interacting IP addresses, ports, class of service and protocol interface.

These techniques allow getting a good view on the connections and switching nodes between the edge router and the access node. Information on the quality of the DSL link may be retrieved from the gateway, e.g. using the TR-069 [6] interface via an Auto Configuration Server (ACS).

However, one domain that has not yet been studied in full detail is how to monitor the actual home network and the connection between the access node and the end device. The access network provider should also extend its monitor reach to this segment of the end-to-end connection and its impact on the QoE of the services, since the cause of QoE degradation is often located here. In a next section, we will discuss how this can be achieved by both active and passive monitoring techniques.

The amount of data that is made available by all these measurements is huge, so the monitor plane should also be able to reduce this amount. A number of techniques exist to summarize large sets of data and to limit the amount of bandwidth required to distribute it. These include using sketches as small space approximations of the available data [7] and placing filters close to the sources to minimize the bandwidth that is required to keep a central platform up-to-date on local measurements [8].

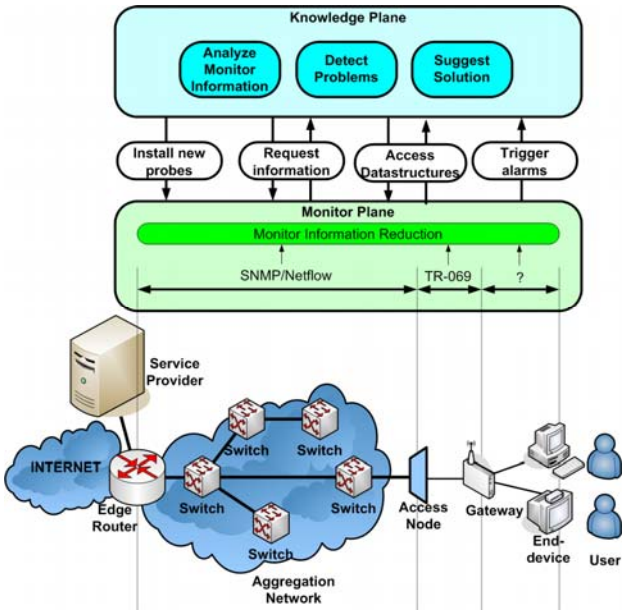

Fig. 2: MP/KP positions in the access network, some technologies that can be used to monitor specific parts of the end-to-end path are also shown.

Knowledge Plane

The knowledge plane is a layer that is built on top of the monitor plane and should be able to analyze and interpret the monitored data. Its primary function is to use all the information that is made available by the monitor plane to

ISBN10 : 907654607X
ISBN13 : 9789076546070

determine if the QoE of some services is degrading and to locate the cause of this service degradation. A second function is then to find a solution to the detected problems.

To perform these functions, the knowledge plane should not only be able to access the available monitor information, but it should also be able to install new monitor probes/modules into the monitor plane and to request additional specific measurements from it (e.g. the counters on a specific switch in the aggregation network). In addition to this, the KP can also set new triggers that generate alarms in the MP. This allows for asynchronous communication between MP and KP. To analyze the monitor data, a collection of techniques and algorithms have already been developed in related projects. These include using deltoids [9] to find measurements that show an abnormal behavior and analyzing the sketches that are offered by the MP to find flows that show anomalous behavior [7].

MP/KP Use Cases for QoE Optimization

A first use case for the MP/KP is related to streaming video based on the RTP/RTCP protocol (Real Time Protocol/ Real Time Control Protocol), where retransmissions can be requested and sent via the mechanisms described in [10] and [11]. We assume that the MP/KP is deployed in the access node. When there is a lossy wireless link in the home network, more retransmissions are requested for lost packets. This effect may, on a larger scale, lead to congestion in the aggregation network. When the MP detects these congestion conditions as well as the increased retransmission requests, the KP is alerted. It could then decide to start the buffering and retransmission of RTP packets on the access node towards the end-device. In doing so, the retransmission can be handled much faster, without generating more traffic in the aggregation network.

A second use case for the MP/KP is for IPTV service. Next to video picture quality, fast channel zapping is a key contributor to the overall IPTV QoE. Due to mis-configurations, channel zapping may be executed through (slow) default IGMP mechanisms. When the MP detects the resulting high zapping latencies, it may alert the KP. The KP could then launch a diagnostic action to identify the root cause, and consequently correct the IGMP proxy configuration enabling fast channel zapping.

In this paper, the further focus lies on the monitor plane and on monitoring the services that are being delivered to the home network, from the access node.

Access and Home Network Service Monitor Techniques

To monitor the services that are provided in a home network and the impact of the home network on the QoE of these services, we need to find a location where all the information for these services is available. The access node is an ideal point to perform this task, since it is the only point in the access network where we can track on one place all data of all services going to and from the home network. Monitoring at this point also enables a first isolation of the location of the problem to the aggregation or home network part by simply comparing the monitored data of the service for both parts.

In the remainder of this paper, we detail how the services in the home network can be monitored. Both active monitoring with ICMP (Internet Control Message Protocol) traffic and passively intercepting TCP (Transmission Control Protocol) and RTP/RTCP packets can give an indication on the QoE of the services that are offered on top of these protocols (fig. 3).

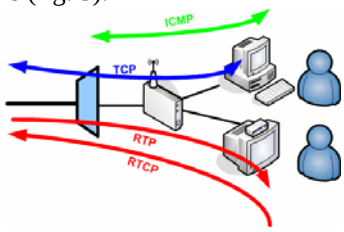

Fig. 3: Access node monitoring and position of the ICMP, TCP and RTP/RTCP protocols

ICMP-based Monitoring

To perform active measurements between the access node and the end device, there are two possible approaches. One can install dedicated software in both these devices, or one can use standard ICMP packets to do these measurements. The first approach makes a huge assumption by requiring dedicated software on the end-device and control of the end-devices by the access network provider. Furthermore, the cost of the end-device might increase substantially when extra software needs to be present. A second approach uses the ICMP protocol. This protocol is supported on every IP device. The ICMP messages, defined in [12], allow sending a message from one IP device to another and receiving a response. Out of this information, one can deduce values for the Round Trip Time (RTT), jitter and packet loss. However, the applicability of ICMP based testing is still hindered by two remaining issues. As the home gateway is very often enhanced with firewall and NAT functionality, it is often not possible to send the packets to the end-device. A second problem is that when you want to measure packet loss, you need to send a significant amount of ICMP packets in order to have an accurate estimate of the path. In [13], a statistical analysis is given that concludes with the observation that in order to measure packet loss ratio of p, $10/p$ packets need to be sent with an interval that is Poisson distributed. Therefore, in order to have accurate estimates of the packet loss, a substantial amount of packets needs to be sent, which might also impact the other traffic and could generate congestion. The authors of [13] have also observed that there is a big difference between the results obtained using active measurements and the counters that are maintained in routers and conclude that great care must be taken in the use of active probes for loss characterization. Because of these disadvantages and since ICMP traffic might also be treated differently than other data traffic and thus can result in inaccurate measurements, we further focus on how to monitor passively the connection between the access node and the end device by looking at the actual traffic for the services.

ISBN10 : 907654607X
ISBN13 : 9789076546070

Monitoring RTP/RTCP Services

Applications like IPTV, Video on Demand, and VOIP often use the real-time transmission protocol (RTP) to send the data to the participants. This protocol is accompanied by the RTP Control Protocol (RTCP), which is used among others to distribute feedback on the quality of the multimedia session to all the participants [14]. Quality feedback is provided by sending Receiver Reports (RR) and Sender Reports (SR), specific types of RTCP messages. An RTCP Sender Report (fig. 4) consists of a report of the sender and a number of reception report blocks containing statistics on received data per source that is contributing to the multimedia flow. The Synchronization Source (SSRC) field in a RTP multimedia session identifies the source of a stream of RTP packets. A receiver report is identical to a sender report, except for the sender-specific fields.

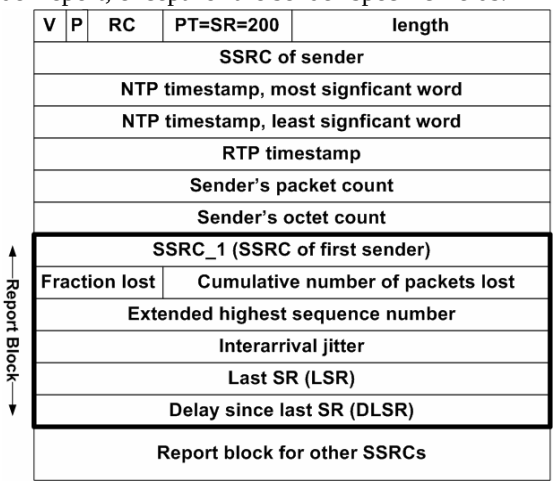

Fig. 4: RTCP Sender report

In order to analyze the connection between the access node and the end-device, we propose to intercept the RTP and RTCP packets in the access node (fig 3). Based on the values that are reported in the RTCP messages and the monitored RTP packets, the following deductions can be made:

- Packet loss: A report block contains values for the fraction of the packets lost and the cumulative number of packets lost since the start of the session. It is also possible to calculate the packet loss of the connection between the access node and the end-device. The field for the fraction of lost packets contains the fraction of lost data packets since the previous SR/RR up until the highest received sequence number. In these fields, duplicate packets are counted twice and packets that arrive too late are not counted as lost. To estimate the packet loss between the access node and the end-device, the access node needs to keep track of the packets that it has sent to the end-device. It can then calculate the fraction of the packets that were lost in the segment between the server and the access node in the interval between two sender or receiver reports. The number of packets lost between the access node and the end-device, in absence of packet duplication, is the difference between the number of packets lost between the server and the end-device (reported in the report block) and the number of packets lost between the server and the access node (counted and observed in the access node).
- RTT: There is no field in the report block to report on the RTT that is experienced by the RTP packets. However, in every report block, the value for the delay since the last received sender report is contained. If the access node keeps track of the time t_1 the last SR passed towards the end-device and also notes the time t_2 the receiver report from the end-device passes it, it can calculate the RTT between the access node and the end-device by computing the difference between these two values, minus the delay since the last sender report (DLSR) reached the end-device: $RTT = t_2-t_1-DLSR$. Although this gives an accurate estimate of the RTT, the frequency is rather low, since the RTT is only computed when RTCP reports are sent.
- Jitter: An estimation of the interarrival jitter between the server and the end-device is reported in the report blocks in the sender and receiver reports and can thus be obtained by sniffing the RTCP packets. The access node can also calculate the interarrival jitter between the server and the access node in the same way as the end-to-end jitter is calculated [14]. These two values can be compared to serve as an indication of jitter between the access node and the end-device.

Apart from these monitoring strategies based on standards RTP [14] implementations, additional monitoring functions can be deployed if retransmissions are used. The IETF is currently working on two drafts, one defining RTP retransmission payload format [11], the other specifying an extended RTP profile for RTCP based feedback [10], defining among others NACK feedback RTCP messages that can be sent out in early (expedited) mode. For a service that makes use of the retransmission approach to ensure a reliable end-to-end transport, we can determine the number of retransmissions by sniffing these NACK messages. By tracking the sequence numbers of the RTP packets that pass the access node, we can estimate the number of retransmissions that are due to packet loss in the home network and on the last mile and those that are caused by packet loss in the aggregation network.

Monitoring TCP connections

TCP, one of the most used protocols on top of IP, is a unicast data transfer protocol that provides reliability and fair bandwidth usage. A number of services, including the basic high speed Internet (HSI) services such as downloading and surfing, use the TCP protocol. TCP is also used for Video on Demand (VoD) services, where the media is played before it has been completely received. This use of TCP is called "Progressive Download". Our goal is to determine the quality of TCP services by sniffing TCP packets on the access node.

Apart from retrieving general home network parameters like jitter and loss, which are interesting for the services running on top of TCP, the TCP service itself can be monitored. TCP adjusts its transfer rate to avoid network congestion, using packet loss as a method to measure congestion. Packet loss is detected indirectly by timeout of

ISBN10 : 907654607X
ISBN13 : 9789076546070

acknowledgment packets. Loss and excessive jitter cause unneeded timeouts and disrupt the progressive download service since TCP will automatically down-scale its sending rate. Such unneeded reduced download speed also needlessly degrades the high speed Internet service (longer downloads, slower web access, etc.).

When only monitoring in the middle of the connection, we cannot directly monitor jitter in TCP, as the TCP header contains no data about jitter. This means we will have to monitor the RTT, and calculate “round trip jitter” from it. Monitoring RTT between client and server in an intermediate point is a non-trivial problem, and several papers have proposed techniques to do this. [15] Outlines a simple method that measures RTT in the 3-way handshake and slow-start phase, at the beginning of every TCP connection. [16] Expands this method, and introduces the “frequency algorithm”, which can measure RTT continuously, by using only the packets going from sender to client. It is based on the fact that inter packet times are approximately periodic, with a period roughly equal to the RTT. While measuring jitter and RTT between client and server is useful for measuring global TCP service quality, it is also interesting to measure jitter and RTT only for the home network part of the TCP connection. This gives a view of how the home network impacts the service quality. Compared to measuring RTT between client and server, this is easy at first sight, as we can simply calculate the time between a data packet passing the access node and the acknowledgment packet sent in reply to it by the client. [17] studies RTT measured close to the end-points of a connection, and so measures RTT in this way. The simple method can find such matching pairs as long as data packets aren’t being retransmitted. This means that once loss and jitter cause retransmissions and duplicate acknowledgments, this method doesn’t find matching pairs, and thus temporarily doesn’t generate measurements. As a result, if jitter occurs, the RTT measurements that can detect it might be missing. This method is thus unfit for reliable jitter detection.

It is also interesting to measure the bandwidth used by each TCP connection. We can measure the “L4 bandwidth”, the amount of data actually being received, or the "L2 bandwidth", the amount of data on the network, which includes lost packets and acknowledgment packets. The easiest way to measure data bandwidth is by looking only at the acknowledgment sequence numbers. These represent up to which byte the receiver has received. Packet bandwidth is also trivial, we simple add up the length of all packets that pass the measurement point.

Packet loss is among the most interesting things to measure since it has a large impact on most other services (such as for example visible artefacts or missing frames in streaming video over RTP). It is typically a symptom of congestion or a poor wireless link. It cannot be measured directly, though the amount of retransmissions can be used as an upper limit to the packet loss. Lost packets will be retransmitted by the reliable TCP. Detection of packet loss in TCP is based on timeout of acknowledgments. This means that not all retransmissions are caused by packet loss, as spurious timeouts, caused by jitter, can cause retransmissions without packet loss. Tests done on the Internet in [18] show that this difference can be as high as 100%.

Retransmissions can be measured fairly accurately and easily. It is sufficient to store the highest seen TCP sequence number and the IP ID of the packet it was seen in. When a TCP packet with a lower or equal TCP sequence number arrives, a retransmission or a reordering has occurred somewhere along the path of the packet. In case of a reorder, the IP ID will be lower than the one of the packet with the highest seen sequence number and in case of a retransmit it will be higher. This is not defined explicitly in the IP RFC, but in practice this is always the case. There is one scenario where this simple method fails: if the last packet (or packets) of a TCP window was lost before the measurement point. This makes the method less useful for small file transfers.

It is also possible to estimate packet loss more directly. Several papers on this topic exist. In [19] for example, an algorithm is presented that discards suspicious retransmissions to estimate an accurate loss rate.

Monitoring IPTV services

The access node monitoring mechanisms described in the previous paragraphs complement the service assurance portfolio for IPTV services. This portfolio will further include, among others, end-to-end monitoring at the Set Top Box as well as base-band video monitoring at the video head-end. The former may for example log video picture quality statistics taking into the packet loss concealment capabilities of the STB. The latter may for example detect black screen and missing audio impairments. Another valuable entry for monitoring is the IPTV middleware platform which may for example track DRM issues.

TCP Connection Tracking Performance

In order to do TCP monitoring, we need to distinguish all connections and to store data for each connection. Processing the interesting monitoring data can require minimal overhead, like in the case of retransmission monitoring. This means that the connection tracking mechanism itself might be an important bottleneck. For this reason we will evaluate the performance of connection tracking. We have tested the performance of Netfilter [20] connection tracking, since the Netfilter implementation is very mature and efficient, and it is quite straight forward to add monitoring code to Netfilter. We have tested the performance of a router both with and without connection tracking. The machine used has two Dual Core AMD Opteron 270 Processor and 4GB of RAM. The on-board interfaces of our test machine where connected internally to the PCI-express bus, which offers a bandwidth beyond 2Gigabit, and thus forms no bottleneck (like a normal 32-bit PCI bus would). Since our machine's double CPUs (each with 2 cores) combined were powerful enough to manage all traffic, we disabled SMP in the Linux kernel. This way, we only used a single core on a single CPU. We also changed the CPU's speed by using frequency scaling, we ran the tests at 1000MHz, and at 2000MHz. 2 kernels where

ISBN10 : 907654607X
ISBN13 : 9789076546070

used, one with Netfilter and connection tracking built in, and the other without. Since the processing power is required per packet and not per byte, we have sent 76 byte TCP packets, in order to maximize the number of packets over a 1Gigabit/s link.

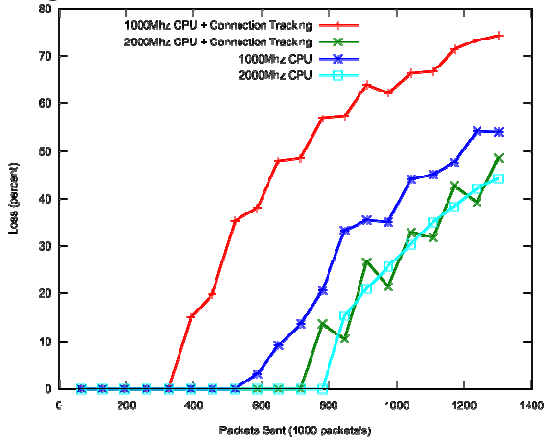

Fig. 5: Loss rate vs. load

Figure 5 shows the result of our tests. At 1000MHz, the connection tracking router starts dropping packets at 32% of 1Gigabit/s (1.6 million packets/sec), while the router without connection tracking starts dropping packets at 39% (2.6 million packets/sec). At 2000MHz, we get respectively 58% (3.7 million packets/s) and 60% (4 million packets/second). The results show that with a modest CPU, one is able to track data flows of millions of packets. The overhead incurred by connection tracking is acceptable. Therefore these results indicate that the approach of tracking and analyzing TCP and RTP/RTCP connections at the access node is feasible.

Conclusions and Future Work

In this paper we have outlined the concept of using a monitor plane and a knowledge plane in the access network to optimize the QoE for a variety of services that are being delivered to the end-users. We discussed how the QoE for services can be guaranteed by monitoring at the access node. Some results were presented for TCP connection tracking which show that monitoring at the access node incurs minimal overhead. We are currently fine-tuning and evaluating a new algorithm for measuring TCP loss and RTT in the home network.

Acknowledgments

The research performed for this paper is part of the MUSE project, which is partially funded by the European Commission within IST FP6. The authors would like to thank the MUSE partners for valuable contributions and feedback.

Pieter Simoens and Filip De Turck are affiliated as respectively a research assistant and a postdoctoral researcher at the Fund for Scientific Research (F.W.O.-V).

References

1. D.D. Clarck, C. Partridge, J. C. Ramming, J. T. Wroclawski, “A Knowledge Plane for the Internet”, Sigcomm '03, August 25-29, 2003, Karlsruhe, Germany
2. Multi Service Access Everywhere, http://www.ist-muse.org
3. J. Case, M. Fedor, M. Schoffstall, J. Davin, rfc 1157, “A Simple Network Management Protocol (SNMP)”
4. IETF Remote Network Monitoring working group rmonmib, http://www.ietf.org/html.charters/rmonmib-charter.html
5. IETF IP flow information export working group (IPFIX), http://www.ietf.org/html.charters/ipfix-charter.html
6. DSLForum, Technical report 069, “CPE WAN management protocol”
7. B. Krishnamurthy, S. Sen, Y. Zhang, Y. Chen, “Sketch-based Change Detection: Methods, Evaluation and Applications”, IMC '03, October 27-29, 2003, Miami Beach, Florida, USA
8. C. Olston, J. Jiang, J. Widom, “Adaptive Filters for Continuous Queries over Distributed Data Streams”, SIGMOD'03, San Diego, California, june 2003, pp. 563-574
9. G. Cormode, S. Muthukrishnan, “What's New: Finding Significant Differences in Network Data Streams”, INFOCOM '04, March 7-11, 2004, Hong Kong, China
10. J. Ott, S. Wenger, N. Sato, C. Burmeist, J. Rey, draft-ietf-avt-rtcp-feedback, “Extended RTP Profile for RTCP-based Feedback (RTP/AVPF) ”
11. J. Rey, D. Leon, A. Miyazaki, V. Varsa, R. Hakenberg, draft-ietf-avt-rtp-retransmission, “RTP Retransmission Payload Format”
12. J. Postel, rfc 792, “Internet Control Message Protocol”, 1981
13. P. Barford, J. Sommers, “Comparing Probe- and Router-Based Packet-Loss Measurement”, IEEE Internet Computing, pp. 50-56, September-October, 2004
14. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, rfc 3550, “RTP: a transport protocol for real-time applications”
15. H. Jiang, C. Dovrolis, “Passive Estimation of TCP Round-Trip Times”, Sigcomm 2002, August 19-23 2002, Pittsburgh PA
16. R. Lance, I. Frommer, “Round-Trip Time Inference Via Passive Monitoring”, ACM SIGMETRICS Performance Evaluation Review, Volume 33, Issue 3 (December 2005), pp. 32-38
17. J. Aikat, J. Kaur, F. D. Smith, K. Jeffay, “Variability in TCP Round-trip Times”, Sigcomm '03, August 25-29, 2003, Karlsruhe, Germany
18. M. Allman, W. Eddy, S. Ostermann, "Estimating Loss Rates With TCP," ACM Performance Evaluation Review, 31(3), December 2003
19. P. Benko, A. Veres, “A Passive Method for Estimating End-to-End TCP Packet Loss”, Globecom 2002
20. Netfilter, http://www.netfilter.org/