IS1
3005

RTP connection monitoring for enabling autonomous access network QoS management

Pieter Simoens[1], Bart De Vleeschauwer[1], Wim Vandemeerssche[1]
Filip De Turck[1], Bart Dhoedt[1], Piet Demeester[1]

Tom Van Caeneghem[2], Kris Struyve[2], Hans Dequeker[2], Edith Gilon[2]

[1] IBBT – IBCN, Dpt. of Information Technology, Ghent University, Belgium
Tel: +32 93314980, Fax: +32 3314899, E-mail: Pieter.Simoens@intec.ugent.be

[2] Alcatel-Lucent Bell N.V., Research & Innovation Department, Antwerp, Belgium

The access network has become an infrastructure for multimedia service delivery to residential users. These services include Broadcast TV (BCTV), online gaming, VoIP and video conferencing and are much more sensitive for packet loss, delay and jitter than traditional IP services such as web-browsing and email. We are developing an autonomic framework for QoS management in the access network that consists of a monitor plane for gathering QoS related monitor information and a knowledge plane for problem detection and solution. In this paper we describe algorithms that allow to detect QoS problems for multicast RTP-based services. We also present an architecture that addresses the detection of new connections and can identify QoS problems quickly in a scalable way.

1. Introduction

The access network has evolved from a best effort Internet data packet forwarder to a QoS-aware real-time framework for triple-play service delivery. The new multimedia services such as IPTV, online gaming and VoIP pose new challenges for the access network architecture to satisfy the high user expectations and fulfil the service demands. They do not only require a substantial amount of bandwidth, but are also particularly sensitive to packet loss, delay and jitter.

In order to meet the increasing QoS requirements for packet loss, delay and jitter, an extensive management of the access network is required. This QoS management is complicated by several factors. First, the impact of network impairments depends highly on the affected service. As an example, the QoS of a Video on Demand (VoD) service is less affected by a small delay than a gaming application. While the VoD user only notices a small delay to fill the buffer at start-up, the gaming experience will be continuously degraded. Furthermore, due to the wide range of user devices, the DSL connection and the possible presence of wireless links in the home network, the QoS for the same service can differ drastically between users.

We are studying the development of an autonomic framework to master the complexity of the access network QoS management. This framework must be able to detect QoS degradations, to determine autonomously the root cause of the problem and to execute the appropriate restoration actions. To this end, the autonomic framework is composed of a monitor and knowledge plane. The monitor plane consists of monitoring probes deployed across the access network and acquires an accurate view on the QoS of the running services. Based on this monitor information, the knowledge plane determines the location and cause of possible QoS problems and determines a fitting solution.

Dependent on the business model of the service provider, one cannot assume full remote control and access to the end-device (such as set-top boxes and desktop PCs). In order to extend the reach of the monitor plane up to the end-device, novel probes must be developed. These must monitor the impact of the home network on the QoS while running at an intermediate point that is under control of the service provider, such as an access network element or a residential gateway.

This paper describes monitor algorithms for large-scale multicast multimedia services based on the Real-Time Transmission Protocol (RTP) [1]. Our algorithms allow to assess the individual contributions on the end-to-end QoS of the parts demarcated by the intermediate monitoring point. The remainder of this paper is structured as follows. First, the broader perspective of our autonomous framework is presented in section 2. Section 3 elaborates on the monitoring algorithms for RTP-based multimedia services. Architectural and implementation details are given in section 4.

2. Positioning the Knowledge Plane

Originally, the Knowledge Plane concept was presented as an enhanced network that enables the automatic detection and recovery of faults [2]. We apply this paradigm to the access network and propose to use a double-layered approach to achieve the expected behavior, consisting of a monitor and a knowledge plane. The complete architecture was described in [3]. The autonomic access node was detailed, with examples for QoS restoration and optimization actions.

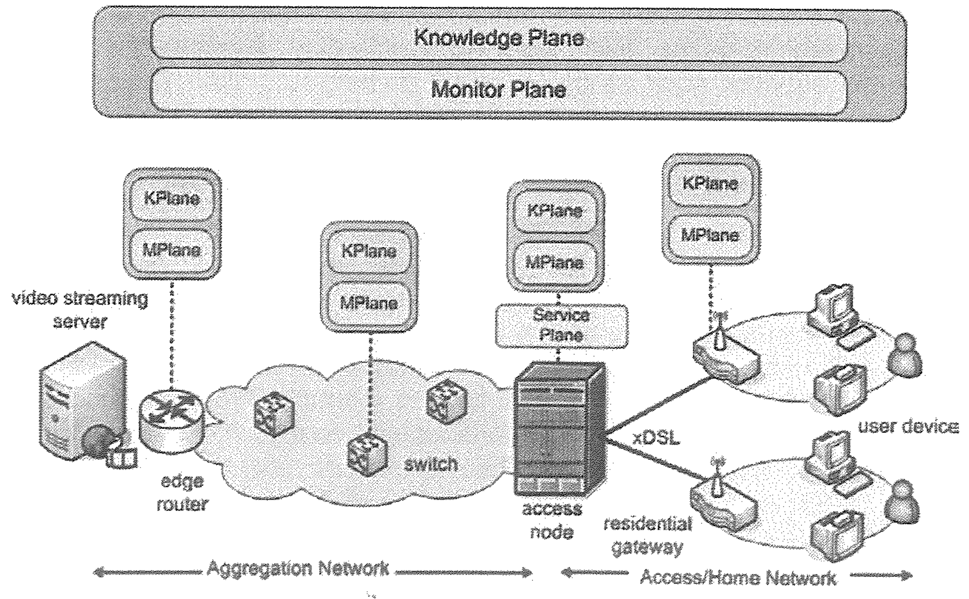

Figure 1 – The Monitor and Knowledge Plane architecture for the access network

The autonomic QoS management framework consists of two layers or "planes'", spanning the whole aggregation and access/home network part from service edge router up to and including the end-device. The first layer, the Knowledge Plane (KPlane), analyzes the QoS of all running services and can restore degraded services autonomously by determining and executing the appropriate actions. The KPlane reasons on an extensive knowledge base with monitor data on the network status. This data is provided by the layer beneath, the Monitor Plane (MPlane). It monitors the status of the network and devices, as well as the QoE of the running services. Each layer consists of a central component and distributed entities in the different parts of the access network, as shown in Figure 1.

To acquire an accurate view on the service QoS, the MPlane must span all elements of the end-to-end path between service edge router and the end-device. However, due to firewall and NAT mechanisms and the lack of a common standard for the user devices, it is difficult to install monitoring tools in the home network. As the access node separates the aggregation network from the access and home network, it is the point nearest to the end-user that can be assumed to be under full control of the access network provider. Furthermore, the access node is evolving to an intelligent service-rich access node by the deployment of a Service Plane [4]. This conceptual layer allows fast deployment of services on the access node in a cost-effective and scalable way. The Service Plane offers micro flow control, enabling a fine-grained service monitoring. Therefore, the access node has a key role in our architecture.

Our current research is devoted to the development of Monitor Plane probes that run on the access node, fueled by the Service Plane. By sniffing both the up-and-downstream packets of protocols with two-way traffic, these probes gather information about the end-to-end status from an intermediate point. In [5], first ideas around a TCP and RTP/RTCP monitor algorithm were presented. The TCP algorithm was fully described in [6], while this paper focuses on the RTP/RTCP algorithm. Since the monitoring algorithms do not rely on the device specifications, they can be applied to all devices adhering to the protocol specifications.

3. RTP/RTCP monitoring

Applications such as IPTV, Video on Demand, and VOIP often use the real-time transmission protocol (RTP [1]) to send the data to the participants. This protocol is accompanied by the RTP Control Protocol (RTCP), which is used to exchange session control information between the participants. A specific type of RTCP messages, the Receiver Reports (RR) and Sender Reports (SR), are used to exchange feedback on the perceived quality of the multimedia session. These reports are sent out at regular intervals by every participant in the session. The minimum interval between two reports of the same participant, is 5 seconds. To guarantee scalability, the interval increases as more participants join the session. Because of this, the total required bandwidth for the exchange of RTCP messages generated by all participants is independent of the number of participants.

An RTCP Sender Report (Figure 2) is composed of two parts. The first part contains information about the data that has been sent out by the originator of the SR. The second part is composed of several report blocks with statistics on received data for each source that is contributing to the multimedia flow. Each participant in the RTP multimedia session has a unique Synchronization Source (SSRC) identifier. A receiver report only contains report blocks and no sender-specific information.

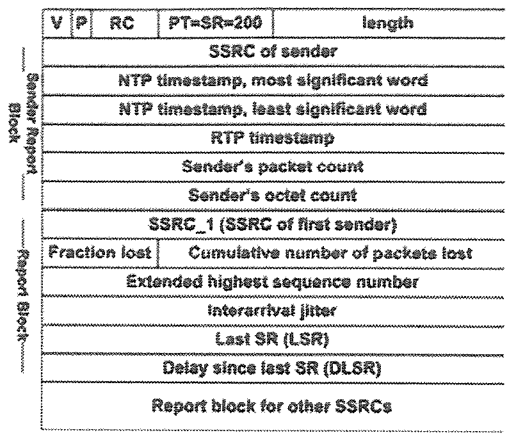
V
P
RC
PT=SR=200
length
SSRC of sender
NTP timestamp, most significant word
NTP timestamp, least significant word
RTP timestamp
Sender's packet count
Sender's octet count
SSRC_1 (SSRC of first sender)
Fraction lost
Cumulative number of packets lost
Extended highest sequence number
Interarrival jitter
Last SR (LSR)
Delay since last SR (DLSR)
Report block for other SSRCs
Sender Report Block
Report Block

Figure 2 - 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. Based on the values that are reported in the RTCP messages and the monitored RTP packets, information about the packet loss, roundtrip time and jitter in the different parts of the connection can be deduced.

3.1 Packet loss

A report block contains values for the fraction and the cumulative number of packets lost since the start of the session. The reported fraction is calculated as the number of lost packets in the interval between the moment when the previous sender/receiver report was generated and the reception of the packet with the sequence number reported in the highest sequence number field. As consecutive packets have increasing sequence numbers, packet loss can easily be detected.

By sniffing the reports on the monitoring node, information on the total packet loss between server and client can be retrieved. To determine the packet loss between server and monitoring node, the client loss algorithm is mimicked on the node. By comparing the reported loss with the loss measured on the node at the moment the last packet of the measurement interval passed by, the loss on the parts demarcated by the monitoring node can be determined.

Several factors can degrade the accuracy of this algorithm. First, there are some inaccuracies in the lost fraction field reported by the client due to retransmitted packets. Furthermore, packets arriving too late at the client are not counted as lost. However, tests have shown that these factors only have a minor impact on the accuracy [7].

3.2 Roundtrip Time

Every report block contains the delay between the last SR was received by the client and the moment the current report block was generated (DLSR). If the access node keeps track of the moment t_1 that the last SR passed the access node and also tracks the moment t_2 when the receiver report sent out by the end-device passes by, 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 reached the end-device: $RTT = t_2\text{-}t_1\text{-}DLSR$. Although this gives an accurate estimate of the RTT, the measurements can only be done with a rather low frequency, since the RTT is only computed when RTCP reports are sent. The interval between two RTCP reports depends on the number of participants in the session, but is minimal 5 seconds.

The one-way delay between server and access node can be determined by sniffing the timestamps of passing RTP packets. This algorithm requires synchronized clocks on server and monitoring node. To determine the RTT between monitor node and client, the clocks must be synchronized with the client clock.

3.3 Jitter

An estimation of the inter-arrival 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 inter-arrival jitter between the server and the access node in the same way as the end-to-end jitter is calculated [1]. These two values can be compared and serve as an indication of jitter between the access node and the end-device. However, since the jitter calculation is not linear, the accuracy of this method is limited [7].

4. Architecture

When monitoring RTP/RTCP connections, several challenges must be addressed.

- *Packet access*. Since many packets pass the access node, we need techniques that allow to receive (only) the packets of interest and to pass them to the relevant handlers.
- *Connection detection*. When a home user starts a new service, e.g. zapping to a new channel or starting a VoD progressive download, the access node needs to detect the presence of this new connection.
- *Scalable monitoring*. Since there may be many subscribers to one multicast RTP stream (e.g. in the BCTV case), it would be useless to monitor the packets that are shared by all these subscribers for each user individually. The architecture must enable to monitor information that originates at the video server once and link this information with client-specific monitor information.
- *Management of monitor node resources.* When there are many connections going through one access node, it is not possible to monitor all of these all the time due to resource constraints. Therefore, the architecture must incorporate decision algorithms for the choice of monitored connection.

These challenges are addressed by the different components of the RTP monitoring architecture, presented in Figure 3. This is detailed in the next paragraphs.

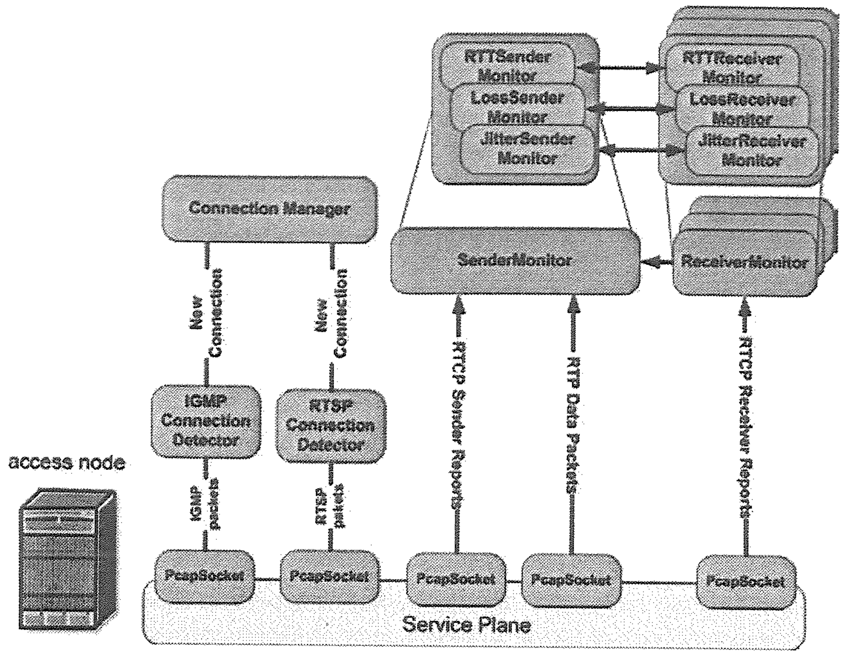

Figure 3 - RTP monitoring architecture with connection detection

4.1 Packet passing

We use a two-layer approach for packet interception. Packets are intercepted with a pcap-like technique [8]. The whole interface to the underlying layer is represented via the *PcapSocket* class. The *PcapSocket* can be equipped with specific filter rules for the kernel to only forward the relevant packets (e.g. based on destination IP address or UDP port). When monitoring an RTP connection, a *PcapSocket* will be used for the RTP data packets from server to client, one for the RTCP traffic from server to client and one for the RTCP packets from the client to the server.

Other classes can register as a listener to the *PcapSocket* to receive the packets. When a connection is detected, the connection manager will instantiate the relevant *PcapSocket* and also register the appropriate listeners with the *PcapSocket.*

4.2 Connection detection

There are several approaches that can be taken to determine when new connections are set-up by clients behind the access node:

1. *IGMP based connection detection:* By intercepting the IGMP join messages, the access node can detect when a new connection is set-up. The corresponding IGMP leave message can be used to determine when a client is no longer viewing a channel.
2. *RTSP based connection detection:* When a client subscribes to an RTP session via the RTSP protocol, it is possible to retrieve information on the client and server IP addresses and ports from the RTSP SETUP messages. In this way new connections can be identified.
3. *IP based connection detection:* By monitoring the traffic from/to a specific IP address, we are able to identify whether a client is receiving a new multimedia stream. The corresponding connection detector could have a list of IP addresses for the multimedia servers in the access network.

4.3 Scalable monitoring

The packets that are sent during a multicast session can be divided into two subclasses: packets that can be used to deduce QoS information for all the subscribers, and packets that are specifically related to one user. Therefore, the algorithms are split into two general classes: *SenderMonitors* and *ReceiverMonitors*. A SenderMonitor receives the header information from the RTP data packets and the RTCP Sender Reports. It keeps this information on a stack and determines the values for a QoS metric between the Video server and the monitor node. A *SenderMonitor* also makes the monitor information available for several ReceiverMonitors. Each ReceiverMonitor is coupled with one user participating in a specific RTP session. They receive the RTCP Receiver Reports and request additional information from the SenderMonitor corresponding to the session. The ReceiverMonitor combines the end-to-end information from the receiver report and the information obtained from the SenderMonitor to determine the QoS of the connection between the access node and the end-device. In the architecture there are separate monitors for each QoS metric.

4.4 Measurements

Each SenderMonitor instance has a stack on which a data structure with information is pushed for each passed RTP packet. When a Receiver Report is received, the stack can be cleaned up to the data structure corresponding to the packet with sequence number equal to the minimum of all reported "highest sequence number" fields. If only one client has joined the session, the stack can be cleaned completely. When more clients join the session, the stack can only be cleaned partially, resulting in a slightly higher average stack size. However, in the multi-client case the stack is cleaned more frequently, resulting in a lower maximum stack size. Table 1 presents measurement results, averaged over 30000 packets passing the monitor node. Since the maximal stack size for monitoring one connection is only around 5.6 kilobyte and the required memory does not increase with the number of subscribers, the memory requirements for monitoring RTP channels are very low.

Table 2 shows time measurements of the period between a RR is received and the loss, RTT and jitter values are updated. This is almost instantaneously. The measurements are performed on a standard Linux desktop, with an AMD Athlon 64 2 GHz processor and 512 MB RAM.

Table 1 - Stack size measurements (kB)

	Maximum	Average	Deviation
1 client	5.628	1.234	0.780
2 clients	5.436	1.271	0.869

Table 2 – Time measurements (μs)

Average	Minimum	Maximum
168	15	1285

5. Future work

As the threshold of tolerable packet loss, delay and jitter can vary on a per-user and per-service basis, a fine-grained mechanism must be installed to track all running services individually. On one hand, continuously monitoring all connections would generate a huge load and would require a lot of processing power. On the other hand, possible QoS problems must be detected rapidly, so each active connection must be checked at a sufficient high rate. Therefore, a decision algorithm is required that strikes a good balance between resource consumption, scalability and accuracy. This decision algorithm will be implemented in the Connection Manager, shown in Figure 3. We are currently looking into a round-robin approach, whereby each active connection is monitored during some period. The frequency of monitoring is determined by a weight factor, which can be a combination of static and dynamic factors.

Another research track will be oriented towards the implementation of restoration and optimization actions in the Knowledge Plane.

6. Conclusion

A double-layered autonomic framework can help to tackle the complex task of QoS service management in the access network. The lowest layer, the Monitor Plane, consists of monitor probes across the access network. For some business models, one cannot assume full access to the home network devices. Therefore, we are developing probes to track packet loss, delay and jitter in the home network that run on the access node. The probes for the RTP/RTCP protocol correlate the information in the Sender and Receiver feedback reports with the RTP packets sniffed at the access node.

In this paper, we presented the monitor algorithms of the RTP/RTCP probes. The architectural design tackles challenges like connection detection, packet passing, managing monitor node resources and scalability. We are currently studying how a round-robin scheduling algorithm that takes into account static and dynamic factors to select the monitored connections, can combine both scalability and accuracy.

Acknowledgements

Pieter Simoens is funded by PhD grant of the Fund for Scientific Research, Flanders (FWO-V). Filip De Turck is funded by postdoctoral grant of the Fund for Scientific Research, Flanders (FWO-V). This research is performed within the framework of MUSE [9], funded by the European Commission as part of the IST 6th European framework.

References

[1] H. Schulzrinne et al., "RTP: A Transport Protocol for Real-Time applications". *RFC3550*

[2] D. Clark et al., "A Knowledge Plane for the Internet", *Proc. of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, Karlsruhe (Germany), 2003

[3] P. Simoens et al., "Towards Autonomic Access Networks for Service QoE Optimization", *Modelling Autonomic Communication Environments*, pp. 223-233, Dublin (Ireland), October 2006.

[4] E. Six et al., "Multiservice Access Network with Distributed Service Enablers", *BroadBand Europe*, Geneva (Switzerland), December 2006

[5] B. De Vleeschauwer et al., "On the enhancement of QoE for IPTV Services through Knowledge Plane Deployment", *Broadband Europe*, Geneva (Switzerland), December 2006.

[6] B. De Vleeschauwer et al., "Enabling Autonomic Access Network QoE Management through TCP Monitoring", *accepted for publication in Proc. 1st IEEE Workshop on Autonomic Communications and Management*, Munich (Germany), May 2007.

[7] Advanced Features for Multimedia enabled Access Platforms, MUSE Public Deliverable DB 1.8, December 2006

[8] TCPDump/Libpcap, http://www.tcpdump.org

[9] MUSE – Multi-Service Access Everywhere, http://www.ist-muse.org

Acreo
Research and development
in electronics, optics and
communication technology

acreo

Proceedings

ERICSSON

e-Photon ONe

Next generation Optical networks for Broadband European Leadership phase 2

NOBEL2

Information Society
Technologies

Muse

LASAGNE

MUPBED

NOC 2007 Proceedings

12th European Conference on Networks and Optical Communications

incorporating Papers of

Conference on Optical Cabling and Infrastructure - OC&I

Edited by
D.W. Faulkner, R. Neat and E. Vanin

June 2007

Supported by

Ericsson AB

EU IST FP6 Projects MUSE, LASAGNE, NOBEL, e-PHOTON/One, MUPBED

ISBN 978-91-633-0869-7
Published by Acreo AB, Kista-Stockholm, Sweden