Thesis for a Master of Science degree in Telematics,

Faculty of Electrical Engineering, Math and Computer Science,

Design and Analysis of Communication Systems (DACS),

University of Twente

Fast retransmission for multicast IPTV

MARTIN PRINS

JUNE 27, 2008

Supervising Committee:

Dr. Ir. Georgios Karagiannis	University of Twente, Enschede
Dr. Ir. Aiko Pras	University of Twente, Enschede
Dr. Marcus Brunner	NEC Europe Ltd., Network Laboratories, Heidelberg

Abstract

In a IPTV distribution network, broadcast television channels are distributed using multicast stream delivery. Packet loss occuring during transport will impair the displayed video signal and thus reduces the Quality of Experience. Due to the nature of video compression techniques a single lost packet can lead to visual impairments lasting for multiple seconds, so packet loss should be kept to a minimum.

Two well known error recovery techniques are packet retransmission and Forward Error Correction (FEC). In a large multicast distribution network an end-to-end packet retransmission mechanism is not feasible as feedback implosion will occur when receivers notify the source about what packets they need retransmission of. A FEC mechanism allows the IPTV stream receivers to recover a certain amount of data, but when loss rates vary for different users there will either be some users with remaining losses or bandwidth will be wasted in large parts of the network where the loss rate is low. Another solution is to use local loss recovery for smaller parts of the multicast distribution tree. By introducing a fast-retransmission function in the access network, losses can be recovered rapidly and the video quality for the users can be maintained.

Based on a literature study and company requirements a design of a fast retransmission mechanism is presented, intended for deployment in an access node. For the delivery of the IPTV stream the Real-time Transport Protocol (RTP) is used. Two recent RTP protocol extensions have added functionality for time-constrained feedback and a retransmission payload format, which could be used for a retransmission mechanism mission for RTP streaming sessions. As the protocol extensions do not provide a complete retransmission mechanism, the proposed design incorporates the functionality needed to offer packet retransmissions for a time-constrained multicast IPTV service.

A prototype is implemented which is used to evaluate the effectiveness of the packet retransmission mechanism and used to determine which parameters influence the applicability of the retransmission mechanisms. For this purposes several experiments are performed, which are used to evaluate the performance in a uncongested network with different loss characteristics and a network in which packet loss occurs due to network congestion.

Evaluation of the prototype shows the efficiency of the retransmission mechanism to handle losses and its performance in congested networks.

Acknowledgements

My thanks and gratitude go to all people from whom I have received support while performing this research project. First of all I would like to thank Georgios Karagiannis and Aiko Pras who supervised me on behalf of the DACS chair of the University of Twente. Their questions, feedback and suggestions allowed me to greatly improve the quality of the thesis.

I would like to thank Marcus Brunner, my supervisor at NEC, for his insights and suggestions regarding the research I performed at NEC. With his help and the help of the colleagues from the Network Management group, I was able to conduct my project in a nice, stimulating environment. This gave me the opportunity to learn a lot about conducting research in a Research & Development company and gain a lot of insight in IPTV technologies and it's developments.

A special thanks goes to my parents Rienk Prins and Gabriele Prins-Barth. Their continuous support allowed me to study and follow my interests which finally lead to a pleasant stay in Heidelberg, Germany. I would also like to thank Sara Prins, Christiaan Prins and Vincent van Kooten for their feedback on this thesis.

Finally and most importantly, I would like to thank my family, my girlfriend and my friends for supporting me during the process of finishing this thesis.

Martin Prins
Enschede, June 2008

Contents

Chapter 1

Introduction

1.1 Motivation

The availability of high bandwidth consumer access networks makes it possible to use IP networks for the distribution of television services, that were previously distributed using alternative distribution channels: television and telephony are well known examples. The availability of broadband access network led to an enormous increase in the usage of Internet based multimedia applications: video conferencing, video streaming and Voice over IP (VoIP) . Internet Service Providers are also seeing new opportunities for the implementation of Internet Protocol Television (IPTV) services. The reasons for these developments are numerous: besides being cost effective, IP based television distribution allows for all kinds of new applications:

- A virtually unlimited selection of TV channels due to dynamic usage of bandwidth;
- Provide TV channels in a much higher quality;
- On Demand services;
- Interactive TV.

IPTV services are distributed (streamed) over IP based networks, using transport protocols like the Real-time transport protocol (RTP) [1], allowing low latency, time constraint stream delivery. IPTV applications are highly vulnerable to packet loss. Due to the manner in which video is encoded the loss of a single packet can lead to visual impairments lasting for multiple seconds. Packet loss can thus severely impact the Quality of Experience for the end user and thus must be prevented if possible.

There are two common approaches for providing resiliency against packet loss:

- Add redundant data to recover from packet loss.
  The redundant data can be used by the receiver to recover packets or packet data that has been lost during transport. The redundant data can either be inserted during the encoding process (application layer forward error correction) or during transport (network layer forward error correction). Adding

redundant data is commonly referred to as Forward Error Correction (FEC).

- Use a retransmission mechanism to retransmit lost packets.
  Upon packet loss, a IPTV client asks for retransmission of missing the packet(s), such that the client can receive the data after retransmission.

Although error resiliency techniques have their benefits, they also have some drawbacks:

- In a large IPTV distribution network (a large multicast tree), the usage of FEC might be inefficient, when packet loss occurs only in a small subset of the distribution tree, or when different subtrees suffer from different loss characteristics. For some parts of the network, the FEC protection may be too strong, therefore wasting bandwidth; in other parts of the network the FEC protection may be too weak to offer sufficient recovery. To provide adequate recovery the FEC protection needs to be improved, leading to an increased FEC bandwidth that will affect all users.
- For packet retransmission to be effective an IPTV client needs to buffer packets. Such buffering allows retransmitted packets to be received without being discarded because they arrive too late. This increase in buffer size leads to an increase of the startup delay for the IPTV service. Since users expect a high defree of responsiveness from the IPTV service, this startup delay must be as small as possible. Furthermore, buffering leads to an increased End-to-End delay, between Streaming Server and IPTV client. For linear broadcast TV this End-to-End delay should be small, to avoid global desynchronization (e.g. a program scheduled at 8 PM will start 10 seconds later).

Linear broadcast IPTV is distributed using multicast distribution, which allows for efficient transmission of TV channels to a large selection of users. In a large multicast distribution network applying retransmission between the Source of the TV Channel and the possible thousands of subscribers is not feasible, as the number of retransmission requests might explode, which might overload the Streaming Server.

Therefore in large multicast distribution trees on a global (session) scale retransmission is not feasible or desirable. As an alternative retransmissions may be applied in specific subtrees of a multicast distribution tree. Thereby adaption to local network characteristics becomes possible, without influencing the entire multicast delivery path. In addition, the retransmission functionality needs only be enabled in the parts of the network where packet loss occurs.

This thesis investigates the application of packet retransmission for multicast IPTV broadcast TV, where error resiliency mechanism based on RTP packet retransmission to be used in a multicast IPTV distribution environment.

1.2 Goal

The goal of this thesis is to design, implement and evaluate a packet retransmission mechanism for multicast IPTV distribution which is used to provide packet retransmission based error resiliency in a subtree of the IPTV distribution path.

1.3 Research questions

To achieve the above stated goal, the following research questions are defined:

- *What are the effects of packet loss on IPTV streaming applications?*
- *What techniques can be used to provide error resiliency for IPTV streaming applications?*
- *How can fast retransmissions be provided for multicast IPTV stream delivery service using the Real-time Transport Protocol?*
- *How can the effects of error resiliency based on packet retransmission be measured?*
- *What are the parameters that influence the performance of the RTP retransmission mechanism?*
- *For which network conditions can RTP-based packet retransmission be successfully applied as an error resiliency mechanism?*

1.4 Methodology

To get a better understanding of the stated problems the thesis starts with a literature study, investigating:

- IPTV technologies;
- IPTV transport protocols;
- Video encoding techniques;
- Causes and effects of packet loss for IPTV applications;
- Error recovery and resiliency techniques;
- Quality measurement metrics and techniques.

Based on the literature study the requirements for packet retransmission in a subtree of a multicast IPTV distribution path are specified.

The requirements are consecutively used to design and implement a prototype IPTV system, which provides packet retransmissions for packet loss originating in the access network of a multicast IPTV distribution path.

The prototype implementation is tested under different simulated network scenarios to determine the effects of packet retransmissions for an IPTV application and to determine which parameters influence the performance of a packet retransmission mechanism for multicast IPTV. The experiment results are used to evaluate the retransmission functionality and determine if and under which scenarios the application of RTP packet retransmission can be beneficial.

1.5 Intended audience

This thesis is intended for readers with a background in telecommunications and with an interest in IPTV services and network management. Basic knowledge about IP networks and multimedia distribution is assumed, although a lot of IPTV specific concepts will be explained.

1.6 Structure of the report

This document is structured as follows. Chapter 2 provides a background study of the technologies used to provide IPTV services. Furthermore the causes and effects of packet loss are presented and error recovery technologies are discussed. This also explains why packet retransmission can be beneficial for IPTV broadcast TV.

Chapter 3 covers the requirements for a fast retransmission mechanism for RTP based IPTV stream delivery in a multicast distribution network.

In chapter 4 the the design and implementation of a prototype for a packet retransmission for a multicast IPTV service are discussed.

To determine the applicability of the fast retransmission mechanism the prototype will be evaluated, both by means of experiments in a lab setup and by means of a analytical evaluation. This is presented in chapter 5. Finally, the conclusions to the research questions will be given in chapter 6 and some ideas to future research will be presented.

Chapter 2

Background

In this chapter an overview is given of the technologies and techniques relevant to the distribution of IPTV services. Furthermore a brief introduction to video compression techniques is presented to give the reader a better understanding of how packet loss might impact an IPTV service. In the last section the techniques being used to evaluate IPTV services in terms of network and application performance are described. These techniques are used to determine the Quality of Service and Quality of Experience of IPTV services.

The following topics will be discussed:

- IPTV technologies;
- IPTV transport protocols;
- Video encoding techniques;
- Causes and consequences of packet loss;
- Error resiliency techniques;
- Quality measurement metrics and techniques.

2.1 IPTV overview

The acronym IPTV stands for Internet Protocol Television. IPTV is commonly interpreted as 'Television services that are distributed over IP networks'. In literature and also in practice a lot of different definitions of IPTV and IPTV services are used, leading to ambiguous interpretation of IPTV and IPTV services. In this thesis the definition formulated by the ITU-T focus group on IPTV will be used as reference [2]:

> IPTV is defined as multimedia services such as television/video/audio/text/graphics/data delivered over IP based networks managed to provide the required level of QoS/QoE, security, interactivity and reliability.

QoS and QoE are abbreviations of Quality of Service and Quality of Experience respectively, two terms used to describe quality levels of a service. These terms will be further discussed in section 2.10. One important aspect of this definition related to the work described in this thesis is "..*managed to provide the required level of QoS/QoE, security, interactivity and reliability*". By defining that the IPTV services delivered using managed IP based networks leads to a distinction between multimedia services which can be regarded as IPTV services and multimedia services that are commonly regarded as Internet TV.

Currently there are a lot of web-based video services which do not offer managed delivery of multimedia services and do not give any QoS guarantees. For instance, the popular video service YouTube [3] offers user contributed videos on-line, but the delivery of these videos is not controlled or managed by YouTube or a related Service Provider. The video content is retrieved by the consumer using a Internet connection, without any guarantees regarding delivery, latency or availability.

Typical aspects of managed IPTV services are:

IPTV services make use of an end-to-end system or semi-closed network IPTV services are typically offered by one service provider which provides the means of making the IPTV services available: the network infrastructure, access to the (television) content, a decoder or Set Top Box used to access, receive, decode and display the IPTV content. The End-to-End service may also depend on multiple parties, the service stays managed and only accessible when allowed by the service provider(s).

IPTV service availability are geographically bound The availability of the IPTV services depend on the network infrastructure. The services are only offered at the locations where the Service Provider has control of the network infrastructure and the network infrastructure offers sufficient bandwidth for IPTV services.

IPTV services are service provider driven Typically the IPTV subscriber uses services offered by the service provider; the user itself does not offer services. In the future IPTV services offered by subscribers (i.e. user based broadcasting) may become available.

IPTV services make use of access and admission control Before a user can use a IPTV service, authorization is used to check if the user has access rights to the content. Furthermore the service will only be offered / available when there is sufficient bandwidth for the service (if not the service will be rejected). This requires a managed network.

Typical aspects of current multimedia Internet television services[4] are:

The services are open to anyone Anyone can have access to the services, as long as they have the means (an Internet connection) to connect to the Service Provider.

Anyone can become a service provider The content can be offered by anyone. This can thus be a TV station offering an on line stream of the TV channel or an individual creating a video for a small number of users.

There is no admission control Although authorization might be required by some (paid) services, there is no bandwidth reservation for the delivery of the content or admission control based on the available bandwidth. This thus can lead to poor performance of the service due to congestion, which may be caused due to non related Internet usage.

A comparison of IPTV and Internet TV services is presented in table 2.1.

	IPTV	**Internet TV**
Users	Geographically bound Requires IPTV infrastructure	Anyone with Internet access
Distribution network	Closed	Open, Internet
Video formats	MPEG-2 MPEG-4 H.264	Windows Media Flash Video H.264
User equipment	Set Top Box and a TV	PC
Security	Admission control Authentication	Publicly accessible Authentication
Video quality	Comparable to analogue TV High Definition	Depends on service Based on available bandwidth
Costs	Subscription Pay-per-view	Free (ad supported) subscription Pay-per-view
Service example	Deutsche Telekom (Germany) Alice Home TV (Italy) KPN Mine (The Netherlands)	YouTube Uitzending Gemist (The Netherlands) Hulu (United States)

Table 2.1: A comparsion of IPTV and Internet TV services

While there currently still is a clear distinction between IPTV services and Internet TV services, these differences are slowly fading: the convergence of multimedia services, the internet and Television services is leading toward consumer devices, the so called media centers. These devices are connected to a TV and can be used to watch television, view on-line movies and browse the internet as well as use multimedia available on the user's PC. Examples of these upcoming techniques are Apple's AppleTV [5] and Microsoft's Internet TV [6].

2.1.1 Advantages of IPTV television over traditional broadcast TV

The main traditional distribution method for broadcast television uses coaxial cables for the distribution of the television broadcasts. These television broadcasts are analogue and are affected by propagation losses. This traditional form of television is gradually being replaced by distribution over IPTV networks and other methods of digital video broadcasting (DVB) provided either via cable (DVB-C), satellite (DVB-S) or terrestrial (DVB-T). Using IP networks for the distribution of television content has the following benefits:

- A higher quality for the subscriber

  IPTV services can be offered in High Definition (HD) format, giving the IPTV user a high quality TV watching experience, because television content can be offered with a higher level of detail and a higher resolution than traditional television supports (i.e. PAL in Europe, NTSC in the US), or Standard Definition (SD) television, which is used for digital satellite or cable TV. In table 2.2 four common resolutions for SD and HD TV are presented. In figure 2.1 a graphical overview of the resolutions of PAL, NTSC, SD and HD television is presented, which clearly shows that a High Definition signal can provide much more information and thus more detail than current television solutions.

- A higher value for the subscriber

  IPTV allows for services which are not, or only to a certain extend, possible with traditional TV. An example would be pausing live TV and resuming it at a future time instance. Also, a much broader selection of TV channels can be offered. Furthermore, because TV services are distributed digitally, degradation of the video / audio quality due to propagation losses will not occur. Furthermore does the usage of IP networks allow for interactive TV services.

- Cost reduction for the Service Provider

  When TV services are being distributed over IP networks, they can easily be combined in the infrastructure of an internet service provider. When a broadband internet connection is available IPTV services are possible. A common broadband product offering is triple play: one subscription for television, telephony and (broadband) internet access.

IPTV services are typically offered over existing broadband cable and DSL networks or deployed in new optical (GPON) networks, which provide sufficient bandwidth for the delivery of IPTV content. Broadband access networks are a requirement, because IPTV services typically require a large amount of bandwidth.

Definition	Abbreviation	Resolution
Standard Definition	SD	720×576; 720×480
High Definition	HD	1280×720; 1920×1080

Table 2.2: Standard Definition and High Definition video resolutions

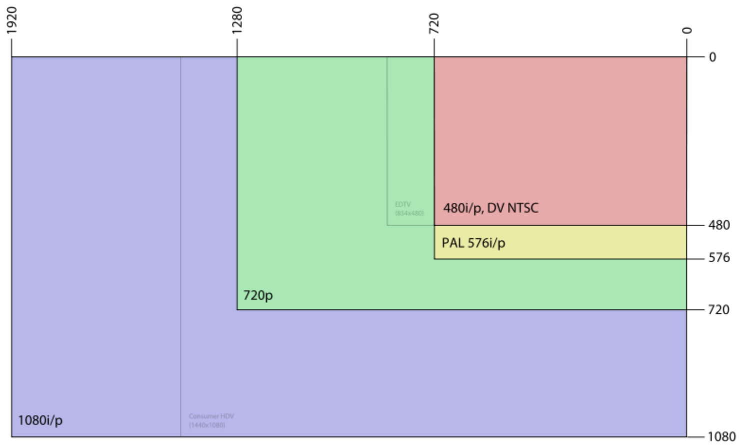

Figure 2.1: An overview of common video resolutions [7]

2.1.2 IPTV services

Typical IPTV services are:

- Linear broadcast television

  Linear broadcast television or live television is the most common form of television: different television stations broadcast their channels via the air, satellite, or cable and the users can select a channel to view the program that the television station is currently broadcasting. IPTV broadcast television is similar to television broadcast currently provided by cable TV or satellite TV. The difference lies in the distribution method: IPTV broadcast television uses multicast IP transport. The subscriber can select from numerous live television broadcasts, witch are being transmitted using multicast delivery.

- Video On Demand

  Video On Demand (VOD) services are interactive television services where the subscriber selects the content and can specify to view the content at a by the user specified time. An example is the rental of a movie, which is commonly known as pay-per-view. VOD services often include trick play functionality: the user can pause playback and can seek in the content.

- Near Video on Demand

  Besides real time VOD also Near Video On Demand (NVOD) services exists. In this case the user cannot exactly determine the playback time: the content is repeatedly scheduled for broadcast. This for instance is used for premium television channels, where the broadcast of a movie starts every hour on a different television channel. But it is also used for IPTV services: in [8] a near video on demand architecture is discussed that combines multicast and unicast delivery, by using scheduled multicast sessions for all users that start watching the program during the scheduled time. Outside the scheduled interval unicast stream delivery is used.

- Time-shifted TV

  Time-shifted TV [9] is a combination of linear broadcast TV and VOD. It provides a flexible viewing window timeframe for television broadcasts, allowing users to watch the beginning of a program, when the broadcast actually already has started. Furthermore, time-shifted TV allows users to pause a live broadcast, to resume it later on. Figure 2.2 shows an example of a Time-Shifted-TV service, allowing users to start watching the show during the 'Start Window' timeframe and allowing users to continue watch the show during the 'View Window' timeframe.

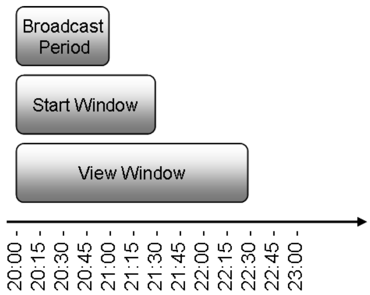

Figure 2.2: A flexible viewing window with time-shifted TV

2.2 IPTV distribution protocols and techniques

For the distribution of IPTV content the audio and video signals must be compressed and digitized. How video compression works is explained in section 2.6. The audio and video streams and optionally other multimedia streams (e.g. subtitles) can be transported separately or combined. The advantage of separate delivery is that it provides a lot of flexibility regarding the distribution of one or more streams. Combined delivery however is less complex as out of band synchronization is not needed. Furthermore does multiplexing lead to a reduced usage of network addresses and ports, an advantage when the number of available (multicast) addressses is limited.

For combined delivery the streams need to be multiplexed and placed in a transport container. A common multiplexing format is the MPEG transport stream (MPEG-TS) format. MPEG-TS provides multiplexing of audio and video and synchronization features of the streams that are transported, such that a receiver can synchronize the streams and can determine when to display the streams. MPEG-TS also provides features for error correction.

When the audio and video streams are not multiplexed, the encoded streams are transmitted directly, without the addition of an transport container or transmitted using a protocol suited for separate stream delivery.

Finally, the packets are then sent using a transport protocol over a IP network to the IPTV user.

2.2.1 IPTV transport protocols

There are different transport protocols that can be used for the delivery of IPTV content. The type of protocol that is or can be used depends on a number of factors. First of all the type of video service is important: live television broadcasts have different requirements than On Demand services. Secondly, when the content is transmitted to multiple users simultaneously some protocols allow for efficient delivery by using broadcasting or multicasting techniques. Finally, the delay or latency requirements of a IPTV application are a important factor to select a suitable protocol.

The following protocols are discussed:

- Transport Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Datagram Congestion Control Protocol (DCCP)
- Microsoft Media Server Protool (MMS)
- Real-time Transport Protocol (RTP)

The first three protocols are real transport protocol. The latter two protocols are not pure transport protocols; they are application layer protocols that run on top of a transport protocol.

Transport Control Protocol

The Transport Control Protocol (TCP) is a reliable connection oriented protocol, which uses a full-duplex connection for the reliable transfer of data [10]. By means of sequence numbers TCP provides in order delivery and a flow control mechanism makes sure that the sender does not send data faster then the receiver can receive and process. A packet retransmission mechanism and a congestion avoidance mechanism allow TCP to provide reliable data transfer and adapt to congestions. This functionality however leads to some constraints regarding the distribution of streaming data: TCP favors reliability over timely delivery. This means that when packet loss occur the receiving application needs to wait before this data is retransmitted, which might lead to buffer underruns. Because TCP adapts to congestion a stable throughput cannot be guaranteed; this means that the receiving application needs to provide a buffer to adapt to the dynamic transfer throughput. Furthermore, TCP requires a three way handshake to setup the connection, which takes time. These last aspects make TCP less suitable for applications that require low latency content delivery and not suitable for applications that prefer the loss of data over high transfer latencies.

User Datagram Protocol

The User Datagram Protocol (UDP) is a connectionless protocol, which only provides limited functionality [11]. It is a connectionless protocol, meaning that there is no active connection between a sender and the receiver. This means that UDP does not provide reliable delivery, flow control, congestion control or adaption of the transfer rate to the capacity of the network or the processing speed of the receiver. For UDP transmissions, the sender determines the transfer rate and is not able to determine if a packet was successfully received by the receiver as there is no transmission control feedback. Because there is no end-to-end connection, UDP can be used to transport data to multiple users simultaneously, using broadcast or multicast mechanisms. Another advantage of UDP is the suitability for low latency data delivery, due to the lack of a connection setup procedure or a reliable transfer mechanisms which contribute to the delay of data transfer and delivery.

Datagram Congestion Control Protocol

The Datagram Congestion Control Protocol (DCCP) is a more recent developed transport protocols, which combines some of the concepts of TCP and UDP: it provides congestion controlled unreliable delivery of unreliable datagrams of over bidirectional unicast connections [12]. DCCP provides a trade off between timeliness (UDP) and (congestion) controlled delivery (TCP), which makes the protocol suitable for applications that have strict timing constraints but can benefit from congestion control. Examples of applications are Voice over IP or video streaming. For these applications the transported data is only valuable in a limited time frame.

Microsoft Media Server

Microsoft's proprietary MMS protocol [13] is a suite of protocols used to stream multimedia from a streaming server to a media player. MMS can use UDP, TCP or RTP for the delivery of the content. The protocol is closed, which resulted that MMS is officially only supported in Microsoft products, but several alternative applications like VLC and Winamp can nowadays also be used to receive media streams that are transported with the MMS protocol.

Real-time Transport Protocol

The Real-time Transport Protocol will be discussed in detail in section 2.3.

2.2.2 IPTV content distribution methods

There are currently four common distribution methods for IPTV services:

1. Unicast distribution

2. Multicast distribution
3. Peer to peer distribution
4. Hybrid distribution

Unicast distribution

For Video-On-Demand services unicast distribution protocols are used: UDP, TCP, RTP, DCCP or for instance Microsoft's proprietary Microsoft Media Server (MMS) protocol are common choices. Because reliable, connection oriented protocols like TCP can introduce high latencies, these protocols are only used for services that do not have low latency requirements. Typically the IPTV user connects to a Streaming Server to retrieve the IPTV content. Once the user is connected the data of the IPTV content is continuously streamed to the user. Prerecorded content can also be transmitted in bursts. In this case the data transfer rate is higher then the application consumption rate. This feature can for instance be used to reduce the startup delay.

For broadcast television unicast distribution is rarely used because of its ineffective usage of the IPTV service provider distribution network: for N users N identical IPTV streams need to be transmitted over the same network.

Multicast distribution

For IPTV services that have many simultaneous users multicast distribution is preferred because this allows for efficient delivery to multiple IPTV clients. An example would be the delivery of live television broadcasts. UPD and RTP are commonly used as transport protocol, but because of the limited functionality of UDP, the Real-time Transport Protocol (RTP) is often used in combination with UDP, because of the specific features for low-latency multimedia content distribution and the availability of a feedback mechanism. A more detailed explanation of the features of RTP is given in section 2.3.

Typically, the TV channels are multicast in the core network and only forwarded in the access network when clients request the respective TV channels. Compared to the core network, the access network has only limited bandwidth capacity. Because an IPTV stream is only forwarded to the user when the user requests the TV channel, an IPTV service provider can offer much more television channels then what is technically possible with analogue broadcast cable TV. A downside of this mechanism is that before the television channel is available for the user, the stream must be requested, whereas with analogue broadcast TV the TV channel is always available in the user's premises.

To enable multicast data transport typically two protocols are used: the Protocol Independent Multicast - Sparse Mode (PIM-SM) [14] and Internet Group Membership Protocol (IGMP) [15]. PIM-SM is a routing protocol for multicast groups; it allows routers to notify each other of available multicast channels and provides multicast routing functionality, including the setup of new multicast distribution path from a source to one or more receivers.

IMGP is a subscription protocol which allows clients to subscribe to multicast groups by means of sending membership reports. Access node routers use these IGMP report messages to determine which users are interested in a certain multicast group (TV channel) and thus to determine if packets from a specific multicast group should be forwarded, and to which router ports.

In figure 2.3 an example of a IPTV distribution network for broadcast TV is given. The TV channels are multicast from the streaming server to the Set Top Boxes (STB), the user equipment which decodes the video stream and displays it on a TV. During transport the stream traverses three networks: the core network, which is maintained by the IPTV service provider; the access network, which connects the user with the service provider and the home network, the network found in the user's premises. The access network and home network are interconnected by a home gateway (HG). The HG is the component which allows devices in the home network, such as a PC or STB, to have connectivity with the outside world. The access network and core network are connected by a Multi Service Access Node (MSAN). The MSAN is a device which integrates different services like television, telephony and internet on one platform and possibly offers connections to different types of access networks. For DSL networks this devices is commonly referred to as a Digital Subscriber Line Access Multiplexer (DSLAM).

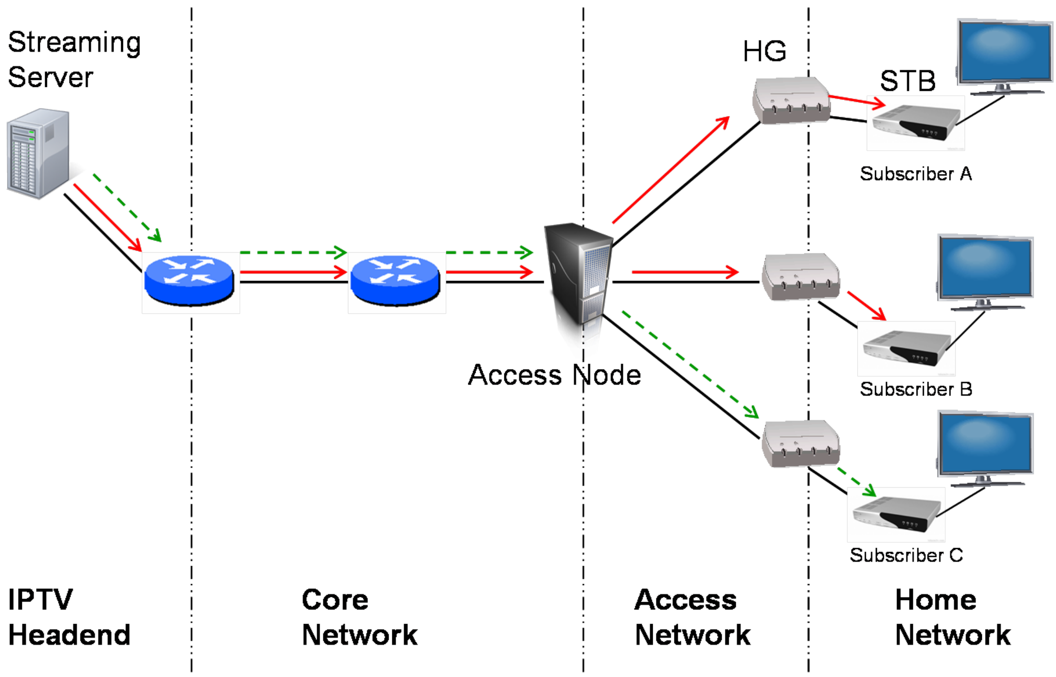

Figure 2.3: A IPTV distribution network for broadcast TV, with two IPTV streams transmitted to different users

Figure 2.3 also shows the distribution of two IPTV streams; one stream is forwarded to subscribers A and B, the other channel is forwarded to subscriber C. When a IPTV user requests a certain TV channel, the STB will issue a request for the respective multicast group by means of a IMGP membership report. This will be received by the home gateway. When the home gateway is already receiving the IPTV packets (for instance when there is a second user in the premises viewing the same channel), the packets are now also forwarded to this user; otherwise it will forward the request to the MSAN. The MSAN will upon reception of the request forward the data from the multicast group to the IPTV client who requested the channel. The IPTV stream will be received by the STB and processed for displaying.

Peer to peer distribution

A relatively new and upcoming technology for the distribution of IPTV services is by using Peer to Peer (P2P) overlay networks to distribute the IPTV content from the Content Provider to all IPTV clients. In a Peer to Peer IPTV distribution network the content is partially or entirely distributed among peers. A client receiving a IPTV stream will not only be consuming the data, but will also be offering (serving) the data to other peers that are interested in the data. From an operator point of view, P2P IPTV is a relatively cheap distribution technique, as the bandwidth required for the distribution of the IPTV content is offered by the participating IPTV nodes and the distribution network is highly scalable.

For peer to peer file distribution mechanisms like Bittorrent [16] peers send and receive data in arbitrary order; it is not important to the user in what order the data is received, as the user will mainly use the file when transfer of the data has finished. For streaming IPTV applications this is however not the case; users would like to start watching a stream as soon as possible and without interruption. This does imply that the order in which the data is transmitted and received between peers is important: a user is only interested in receiving that data that immediately follows the data which is currently being decoded and displayed. For this type of application thus a distribution tree is needed in which nodes in the tree receive data from higher nodes in the tree. This can also mean that a large playback lag may exist between the transmitting node and nodes at the edges of the distribution tree. Furthermore is the dynamic availability of resources very dynamic as IPTV users are constantly joining and leaving the service. To avoid buffer-underruns due to this dynamic behavior a relatively large prebuffer is required. Hei et al. present a measurement study of a large scale P2P IPTV system [17], namely PPLive. PPLive [18] is currently widely used for amongst others the distribution of public Chinese TV channels. The study measurement results show startup delays of 20-30s for popular channels, while impopular channels had startup delays of up to 2 minutes. The measurements also show that playback lag among peers could be as high as 120 seconds. Besides PPLive, other commonly used peer to peer IPTV applications are TvAnts [19] and SopCast [20]. More technical information about P2P IPTV systems can be found in [21] and [22].

Hybrid distribution

Besides the above mentioned distribution methods, hybrid variants are also common nowadays. Hybrid solutions combine distribution methods to optimize content delivery. Two aspects that are often optimized are the startup delay and network distribution costs.

For multicast based IPTV services the startup delay caused by the required IGMP subscription can be reduced by starting to receive the IPTV stream via a unicast connection with a Streaming Server. Via this connection the IPTV client receives the IPTV data as fast as possible. This allows the client to decode and display the TV channel faster than what is possible with multicast distribution. While the content is being displayed the IPTV client joins the multicast group and once the data from the multicast group is received, the client switches from unicast stream to the multicast stream.

The same principle can also be used to reduce the startup delays for peer to peer based IPTV distribution. The Streaming Server then has two purposes: first it allows for small startup delays, secondly it functions as a backup data resource, such that, when there are not enough peers to sustain the stream delivery, a IPTV client can connect to the Streaming Server to receive the missing parts of the stream and keep displaying the IPTV content without interruption.

Streaming versus burst delivery

A typical transport method for multimedia is real time distribution, commonly known as live streaming: the data is transferred or streamed in real-time over the network, thereby minimizing delay. So the data transfer rate resembles the data consume rate.

An approach to reduce the startup delay is to transmit the IPTV data at a rate faster then the consumption or playback rate. By doing so the IPTV client can immediately have a lot of data available at the IPTV client, and therefore requires a shorter period before decoding of the video and audio data can begin. Disadvantages of this technique is that additional buffering delay (and thus playback lag) is introduced in the distribution network. Furthermore, not all access networks have enough bandwidth available to handle burst traffic.

2.3 Realtime Transport Protocol

The Realtime Transport Protocol (RTP) [1] is a transport protocol designed for the transfer of real-time data over the Internet. The RTP protocol was designed to support data with real-time characteristics, to be used for low-latency applications, like telephony, video conferencing, or IPTV. RTP typically runs on top of UDP [11], but other transport protocols like TCP are also supported. RTP itself does not guarantee timely delivery, nor does it provide any reliability, but it provides specific features for streaming multimedia data.

The protocol consists of two parts:

- The transport of realtime data
  This can for instance be an audio or video stream or a combination of multiple streams. For transport a transport layer protocol such as UDP or TCP is used. While support for UDP is mandatory, TCP support is not required.

- Monitoring and signaling of an ongoing transport session
  The monitoring and signaling is provided by the Real-time Control Protocol (RTCP).

What distinguishes RTP from other protocols is that RTP is specifically designed to carry multimedia data: RTP can be used to stream data for low latency applications like VoIP or IPTV. It supports the transmission of multiple streams, allowing for the flexible delivery of separate or combined audio and video streams and its synchronization features allow for flexible streaming scenarios. RTP streams that are for instance transmitted by different sources can be synchronized by a RTP receiver. In figure 2.4 the header of an RTP packet is presented.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.4: RTP packet header

The header has certain fields that make RTP suitable to support (low-latency) multimedia applications:

- **Timestamp** The timestamp field can be used to synchronize multiple RTP streams, determine the scheduled play out time of the payload, and to determine the jitter between sender and receiver.
- **Sequence numbering** The sequence number can be used to detect loss and to reorder packets that are received out of order.
- **Payload Type** This field is used to indicate the payload type of an RTP packet. Currently there are several predefined payload types (see [23], section 6 and [24]) and there are also ranges of dynamic payload types, to be used for data formats that are not yet covered by the predefined payload types list.
- **Synchronization source (SSRC)** The source of a stream of RTP packets. The SSRC field contains a random generated 32-bit (unique) identifier such that all members of a RTP session can determine the source of a RTP stream without depending upon the network address. This is convenient as RTP packets may be combined / mixed during transport. All packets from the same Synchronization source use the same timing and sequence number space, so a RTP receiver groups packets by the SSRC for playback.
- **Contribution Source (CSRC)** When RTP streams from different sources are combined by mixers the receiver can use the CSRC field to determine the source of a packet (as all packets will contain the SSRC from the mixer; the CSRC then tells the source of the packet before it was mixed).

2.3.1 The Real-time control protocol

The Real-time control protocol (RTCP) provides functionality for monitoring RTP sessions, including mechanisms for the identification of the participants in a RTP session and minimal control of the RTP session. For this purpose RTCP provides Sender and Receiver Reports:

- A Sender Report is used by active senders to report about transmission and reception statistics.
- A Receiver Report is used to report reception statistics by a participant that is not actively sending data.

RTCP reports are periodically sent using RTCP packets to all session participants. The total bandwidth usage for RTCP data for all participants is restricted to 5% of the corresponding RTP session bandwidth and a recommended minimum report interval is set to 5 seconds. The 5% upper limit is provided to keep the control data proportional to the data transport; the recommended 5 seconds lower limit is set to avoid RTCP packet floods when a RTP session behaves unexpectedly. By means of the transmission of RTCP reports each participant keeps track of the number of members in a session and can thereby compute it's share of RTCP bandwidth and thus the RTCP report interval. By adaption of the transmission rate to the number of participants RTCP provides a scalable solution for reporting transmission and reception statistics. These RTCP constraints however have implications on the transmission interval for sending RTCP reports: the more members are joining a RTP session the higher the transmission interval between RTCP reports gets. This growth is linear with the group size (such that a constant amount of control traffic is transmitted when summed across all members). For large and very large broadcast groups, the feedback mechanism will therefore become invaluable because the feedback transmission interval will be too high to detect problems and provide a solution.

By the transmission of RTCP reports, problems in RTP streaming sessions can be identified, reported and possibly resolved. For instance, a sender could reduce the transmission rate when a receiver indicates large amounts of packet loss. Another possibility is fault localization in a IPTV distribution network, by comparing the reported loss characteristics from IPTV clients with the characteristics measured in an access node. This principle is further elaborated in the paper by De Vleeschauwer et al. [25].

In figure 2.5 the a RTCP packet containing a receiver report is presented. This receiver report informs the sender (identified by *SSRC_1*) about the packets the receiver (identified by *SSRC*) has received, the fraction of packets that were lost and the inter arrival jitter. Furthermore does the receiver provide the delay since the last sender report, which is used by the sender to determine the round trip time delay between the sender and this receiver.

2.4 Notification and configuration of a streaming session

Before IPTV stream delivery can start it is necessary to inform the receiver about the available streams, setup the delivery of the stream and optionally negotiate streaming session parameters. The exchange

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
header |V=2|P|    RC   |   PT=RR=201   |             length            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     SSRC of packet sender                     |
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
report |                 SSRC_1 (SSRC of first source)                 |
block  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  1    | fraction lost |       cumulative number of packets lost       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |           extended highest sequence number received           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                      interarrival jitter                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         last SR (LSR)                         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                   delay since last SR (DLSR)                  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.5: RTCP packet header with receiver report

of these parameters will typically occur during the setup of a streaming session, or for broadcasting scenarios (e.g. the 24/7 available television channels) will be provided in advance. A common protocol for describing multimedia sessions is the Session Description Protocol (SDP) [26]. A SDP description typically contains media properties (the audio and video codecs used and their settings), transmission properties (the transport protocol used; the network address and ports used) and maybe a description of the content (author, title etc.). To send and receive an SDP description and optionally negotiate transport parameters different protocols are used:

- The Hypertext Transfer Protocol
- The Session Announcement Protocol
- The Real-Time Streaming Protocol
- The Session Initiation Protocol

The Hypertext Transfer Protocol (HTTP) [27] is commonly used for the transfer of data over the world wide web (browsing, downloading, etc.), but it can also be used to periodically retrieve information about streaming sessions. A HTTP client connects to a HTTP server to retrieve information that the server offers, in this case session information (e.g. a SDP file).

The Session Announcement Protocol (SAP) [28] provides the announcement of multicast sessions via multicast. Entities interested in receiving information about the available sessions listen to a well known multicast address to receive information about new or updated sessions which is provided by a SAP announcer. This SAP announcer periodically transmits an announcement packet, containing a (SDP) description of the announced session.

The Real-Time Streaming Protocol (RTSP) [29] is used for establishing and controlling streams of continuous media such as audio and video. The RTSP protocol can be described as a network remote control for Streaming Servers, allowing a user to pause and resume a stream or search in the content. The RTSP message syntax is similar to the Hypertext Transfer Protocol (HTTP) syntax and can be used to request SDP descriptions.

The Session Initiation Protocol (SIP) [30] is a signaling protocol for creating, modifying, and terminating sessions with one or more participants and is more general than RTSP. The SIP protocol is for instance also used for instant messaging, while RTSP is commonly used for streaming applications.

2.5 RTP protocol extensions

The RTP protocol was designed with future extendability in mind: the payload type field allows for providing new audio and video formats; the same holds for the payload type field for RTCP packets. The protocol furthermore specifies how the RTP header can be extended and how new audio and video profiles can be added to RTP.

Over the last years numerous protocol extensions have been proposed and standardized. Some address new functionality for RTP (new audio and video payload types like H.264 or forward error correction [31], or a RTP profile for secure RTP transport [32]) while others address some the shortcomings of the RTP protocols, such as the RTCP transmission constraints for RTP sessions with many participants or a mechanism to provide RTCP reports in a (single source) multicast setup [33], which is a typical setup for broadcast services like IPTV broadcast television.

In the following subsections three new protocol extensions are discussed that extend RTP and RTCP functionality regarding the improvement of the RTCP transmission interval and the retransmission of RTP packets.

2.5.1 Aggregation of RTCP reports

As discussed in section 2.3.1, the RTCP report interval depends on the number of participants in the RTP session; when the number of participants increases the bandwidth per participant decreases, which means that the interval between subsequent RTCP reports from a specific participant gets bigger. Komosny and Novotny have shown that the RTCP mechanism can become invaluable when the group size gets very large [34], [35]. They show that the RTCP report transmission interval is 1963 seconds in a RTP session with 100000 users and a session bandwidth of 1 Mbit/s. A reporting interval of more than halve an hour can be considered too large to provide valuable receiver feedback regarding reception problems (i.e. information will already be outdated). They argue that the transmission interval for RTCP messages in large multicast groups can be decreased if the amount of transmitted messages is decreased. This can be achieved with the aggregation of RTCP receiver reports from different users. By combining receiver reports from multiple users, less bandwidth will be needed to distribute the reports to all RTP participants and thus reduce the RTCP report transmission interval.

They propose a hierarchical architecture for the distribution of RTCP receiver reports. In a tree based structure, the RTP receiver nodes report to summarization nodes, which aggregate the receiver reports and add their own report if they also participate in the RTP session. With this tree-based aggregation all receiver reports are forwarded toward the RTP sender or alternatively a reporting node. As an example the authors mathematically show that the RTCP transmission interval reduces from 1963 seconds to 14 seconds, which is 140 times better. Because of the number of supported nodes and the low RTCP transmission interval, this solution is suitable for providing session feedback in large (IPTV) multicast networks.

Another proposal for RTCP report aggregation does not address the report interval of the reports, but an RTP application scenario in which providing RTCP reports via multicast is not possible or not desirable. Examples are single-source multicast setups, which do not provide the possibility for receivers to send to the multicast group, or network restrictions imposed by a service provider, as multicast data originating from subscribers lead to a high or unbearable load on the IP multicast service in the network. Another reason for restricting subscribers from transmitting multicast data is that it might lead to privacy concerns, as personal data in a RTCP report from for instance an IPTV user may be readable by other IPTV users.

In the proposed standard, "RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback" [33], RTCP reports are transmitted using unicast transmission to a feedback target, which can be used to aggregate reports from different clients. The reports are then redistributed by a distribution source to all participants of a RTP session.

2.5.2 Extended RTP profile for RTCP based feedback

For some applications the delay of the transmission of RTCP reports may be undesirable, for instance when the information contained in the RTCP reports is only valuable for a limited amount of time. An example is the usage of RTCP reports to notify packet loss, which could be used for a retransmission mechanism. In 2006 an extension to the RTP standard was proposed which allows the RTCP protocol to be used for time-constrained feedback by reducing the RTCP report transmission interval [36].

The standard, RFC 4585, specifies a new mechanism to determine when RTCP reports should be transmitted. The lower bound of 5 seconds between successive reports is removed; the interval is only derived from the average RTCP packet size and the RTCP bandwidth share available to the participant. Optionally, a minimum interval between regular RTCP packets may be enforced. Furthermore does the mechanism allow a participant to send a RTCP message earlier then the next scheduled transmission time. This type is called early RTCP mode. When a report is sent in early RTCP mode, the time slot for the next regular RTCP packet is updated accordingly, to ensure that the short-term average RTCP bandwidth used with early feedback does not exceed the bandwidth used without early feedback.

The protocol extension specifies three RTCP transmission modes:

- Immediate Feedback mode
  In Immediate Feedback mode, the group size is below a specific feedback threshold, which gives each RTP receiver enough bandwidth to transmit the RTCP feedback packets for the intended purpose.
- Early RTCP mode
  In Early RTCP mode, the group size (and other session parameters) do not longer allow for each receiver to react to each event which would require reporting. In other words, a receiver will not be able to report all feedback messages because of the protocol constraints, to prevent a high load of RTCP data negatively influencing the streaming session. But RTCP feedback can still be given sufficiently often to allow the session sender to adapt the session media bandwidth accordingly to improve the overall media playback quality.
- Regular RTCP mode
  In Regular RTCP mode, it is no longer useful to provide feedback from individual events from receivers because of the time scale in which the feedback can be provided, and/or in large groups senders are not able to react upon all individual requests from receivers (i.e. process all feedback).

The specific feedback threshold depends on a number of technical parameters (type of codec, type of transport, type of feedback) but also on application scenarios. An additional feedback suppression mechanism makes sure that in multi party sessions feedback implosion does not occur. For time constrained feedback the protocol extension provides two feedback modes: acknowledgement (ACK), which can be used for unicast RTP sessions and negative acknowledgement (NACK), for unicast and multicast sessions.

Besides a new protocol for RTCP transmission, the standard provides packet formats for low-latency RTCP feedback (FB) messages, divided in three categories:

- Transport layer FB messages
- Payload-specific FB messages
- Application layer FB messages

Transport layer FB messages can be used for general purpose feedback at the transport level. A predefined message type is the generic negative acknowledgment (NACK) message. Payload specific FB messages can be used for payload dependent feedback. This can for instance be used to notify about specific video frames that are missing. Application layer FB messages can be used to transparently transmit feedback from the receiver's application to the sender's application.

This protocol extension provides building blocks for creating applications that use RTP and are in the need for low-latency feedback. It does however not specify a complete protocol of how (often) the feedback should be offered or how much bandwidth should be used for the feedback messages; it is still up to the application developer to judge and to decide what is acceptable or recommended.

2.5.3 RTP Retransmission Payload Format

In IETF standard RFC 4588 [37] a RTP retransmission payload format is specified that can be used in combination with the feedback mechanism discussed in the previous paragraph to create a packet loss recovery technique for RTP streaming sessions. The RFC specifies a retransmission payload format and two transmission schemes to provide the retransmissions:

- Session-multiplexing
  Session-multiplexing is based on sending retransmissions using a different RTP session, i.e. an additional RTP session with a different destination address and/or port is created to be used for retransmissions. By having different sessions for 'regular' transport and retransmissions there is a lot of flexibility: a RTP receiver can choose to join the retransmission session and different transport techniques can be combined, for instance a multicast RTP stream with unicast streams for packet retransmissions. This furthermore allows session-multiplexing for differential treatment in the network (i.e. lower the priority of the retransmission stream) and may simplify processing by network components (i.e. packet caches). A potential drawback of this technique is that more network addresses need to be used, which can be problematic when the address range is limited, especially in the case of multicast.
- SSRC-multiplexing
  SSRC-multiplexing is based on using only one RTP session for the normal packets and the retransmission packets. The main advantage of this method is the usage of only one port for transmitting the RTP packets which allows network components that are involved in distributing the RTP streams to minimize port usage.

Both methods can be used for unicast streaming sessions. For multicast streaming Session-multiplexing must be used, because the association of the original stream and the retransmission stream is problematic if SSRC-multiplexing is used with multicast sessions. The motivation for this is described in section 5.3 of the standard.

2.6 Video compression technologies

In this section an introduction into video compression principles and technologies are given. This section is provided to get an understanding of how packet loss effects the video quality of an IPTV stream. It is out of the scope of this thesis to discuss specific video format detail, but the basic concepts are explained and some common IPTV video formats are discussed. Video encoders reduce (compress) the size of a video signal to allow video footage to be stored or distributed using resources with a limited storage or throughput capacity, like a DVD or a broadband internet connection. A video compression format is often addressed as a codec, which is an acronym for *co*mpression/*dec*ompression. To compress video content, video encoders make use of three principles:

Fidelity Fidelity defines the accuracy in which the compressed image reproduces the original image. A video encoder might for instance reduce colour space: similar colours are replaced by a colour that approximates the original colours. The higher the number of colours that will be replaced by one colour, the lower the fidelity, but the higher the compression ratio. An extreme example would be replacing colors with black and white. Another option is reducing the resolution of an image; by reducing the resolution some details are lost, but the resulting storage size can be much smaller. Figure 2.6 shows examples of fidelity based compression by a reduction of the image resolution and reduction of the colour space.

(a) Original

(b) reduced resolution

(c) reduced color space

Figure 2.6: Fidelity compression: (a) original, (b) reduced resolution and (c) reduced color-space

Spatiality Spatiality defines the relation between parts of a image. When an image is divided in smaller blocks, it is likely that neighboring contain the same color, because they belong to the same object presented in the image. If for example an image shows a red balloon and the image is divided in 1000 blocks, it is likely that multiple blocks contain the same information (they for instance have the same color). The video encoder tries to remove this redundant information to save storage space.

Temporality Temporality describes the relation between subsequent video images in a video sequence. A video image from a video sequence is commonly referred to as a video frame. Subsequent video frames tend to (partially) contain the same information: subsequent frames may show the same object or parts of the objects, because the location of the object has changed. Because neighboring frames often have large similarities, a higher degree of compression can be reached by only storing the differences between subsequent frames. The similarities are not encoded and thus the resulting size of a frame that only contains the differences with the previous frame is much smaller than the original frame. Figure 2.7 shows two subsequent frames of a video file. The two frames have a lot of redundancy; only small differences between these frames are visible: the mouth and the left hand of the news reader are the only (easily) noticeable differences between the frames. The other parts of the frames are identical and thus do not need to be stored in both encoded frames.

Figure 2.7: Example of the temporal relation between two subsequent frames

Video encoders typically use different types of compression for different frames: reference frames and predictive frames. Reference frames are standalone images that have no temporal relation with other frames. Predictive frames have a temporal relation with other frames. To improve the level of compression, video streams often have one reference frame per one or two seconds of footage. The frames belonging to a reference frame is commonly referred to as a Group of Pictures (GOP). The interval between two reference frames is called the GOP length.

2.7 Video compression standards

Standardisation is an important aspect for the deployment of video formats: it ensures that several manufacturers can create interoperable solutions allowing a video standard to be used on a large scale, it allows for a reduction of costs and allows for an increase of performance, as experts can contribute to the development of the standard. An example of successful application of video standards is the MPEG-2 standard for DVD video.

There are two organizations focusing on the development of open video and audio coding standards:

- The Moving Pictures Expert Group (MPEG). MPEG is a working group of ISO/IEC [1] in charge of

[1] The International Organization for Standardization (ISO) is the world's largest developer and developer of International Standards. The International Electrotechnical Commission (IEC) is the world's leading organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

the development of audio and video coding standards. It was established in 1988 and since then has developed standards for products such as video CD and MP3 (MPEG-1 standard), Digital Video set-top boxes and DVD (MPEG-2), and the standard for multimedia for the fixed and mobile web (MPEG-4) [38].

- The Video Coding Expert Group (VCEG). VCEG is a working group of the International Telecommunication Union Standardization Sector (ITU-T) focusing on the development of new video coding standards for conversational (e.g. video conferencing, video telephony) and non-conversational (e.g. streaming, broadcast and file download) audial/visual services [39].

In figure 2.8 the evolution of the MPEG and ITU-T video coding standards is given. It also shows two major video standards that were a combined effort of both groups. It is out of the scope of this thesis to discuss all of the video standards, so only the three key formats will be discussed: MPEG-2, MPEG-4 and H.264.

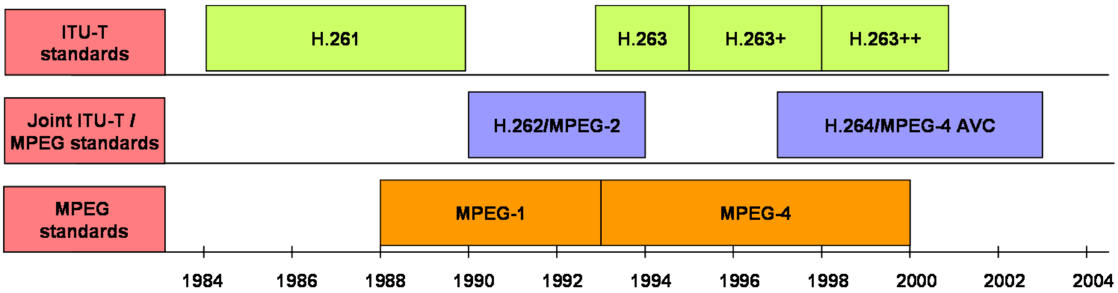

Figure 2.8: Evolution of video coding standards [40]

2.7.1 MPEG-2

MPEG-2 is a video standard developed in the beginning of the nineties by the the Moving Pictures Expert Group. The MPEG-2 video standard is widely used, as it is the video format used for DVD and also a common format for IPTV channels, digital cable TV and satellite TV. MPEG-2 compression uses two types of predictive frames: P-frames or predictive frames are frames that only store the changes with the preceding reference frame. Bi-directional frames or B-frames rely on both previous and subsequent frames. Therefore a higher compression ratio can be achieved. In figure 2.9 an example compression scheme of MPEG-2 is presented, showing

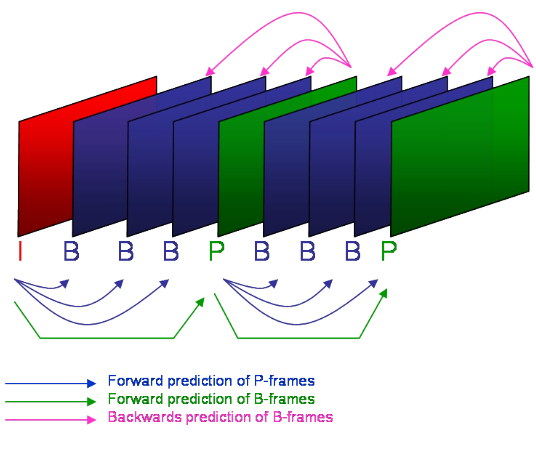

Figure 2.9: MPEG2 compression scheme

the bi-directional relation of B-frames with P- and I-frames and P-frames with the preceding I-frame.

2.7.2 MPEG-4

MPEG-4 is a group of audio and video coding standards for storage and delivery of digital multimedia [41]. Initially the goal of MPEG-4 was to to create a standard for low bit-rate applications as a standard for high bitrate application already existed (i.e. MPEG-2) but this was changed into a standard covering high compression ratios for both low and high bitrates. The standard consists of several sub-standards, covering a group of audio and video coding standards, a framework for rich interactive multimedia, and a standard specifying the storage of MPEG-4 content.

MPEG-4 is the successor to the MPEG-2 standard, extending the application to distribution over (lossy) IP networks, rich media and providing features for interaction. MPEG-4 specifies two different video encoding standards that are currently both being used on a large scale: MPEG-4 Part 2 and MPEG-4 Part 10. When talking about the MPEG-4 format people tend to mean the format described as MPEG-4 Part 2.

MPEG-4 Part 2 or MPEG-4 Advanced Simple Profile (ASP) is a high performance video codec with scalability and error resiliency features. MPEG-4 ASP provides higher compression ratios than MPEG-2 for the same resulting video quality, which resulted in the standard being used for Video On Demand services, online multimedia services, and content delivery to portable handsets. Popular implementations of MPEG-4 Part 2 are DivX [42], the open source Xvid [43] and the Apple QuickTime MPEG codec [44]. Nowadays consumer DVD video players often also support MPEG-4 ASP content which allows consumers to watch a movie that fits on a CD with a similar video quality provided by the same movie in MPEG-2 format on a DVD, which is a large improvement of the compression ratio.

2.7.3 H.264

H.264 is the most advanced video coding standard currently available. The video coding standard is a joint standard created by ITU-T and the MPEG group. The standard is known under different names: Advanced Video Coding, MPEG-4 Part 10 or MPEG-4 AVC as specified by ISO/MPEG and H.264 as specified by the ITU-T. Typically the term H.264 is used to describe this video format.

H.264 is a high performance video codec for demanding applications. It can be used for a large range of applications, including mobile media players (e.g. videos for the Apple iPod), High Definition video content on the next generation multimedia disc (Blu-Ray, HD-DVD) and the distribution of IPTV content. H.264 is likely to replace MPEG-2 as the video standard for IPTV stream delivery, as H.264 provides higher compression ratios at lower storage costs, wich leads to a reduction of the network bandwidth needed for an IPTV stream. This however comes at a price: the processing power requirements for decoding (High Definition) H.264 content are much higher then for MPEG-2.

2.7.4 Layered Video Coding

To provide a TV channel stream for different application scenarios (i.e. in High Definition, in Standard Definition and a stream suited for processing and displaying on a small mobile device), the same content needs to be encoded multiple times, for each application scenario once. This is not an ideal situation as it can contribute to a waste of network and encoding resources when the number of application scenarios increases. A recent development to solve this problem is to use a layered video encoding scheme.

Layered video coding is the technique of encoding a video signal in a low quality or low resolution base layer and optional enhancement layers different layers. The enhancement layers provide quality improvements to the base layer, for example by increasing the resolution or doubling the frame rate, but are not required to be able to decode the base layer video stream. This principle can be used to provide separate streams for the base and enhancement layers, which can be separately transmitted to a IPTV client, based on the available bandwidth and the capabilities of the IPTV client device. When congestion occurs, the transmission of one or more enhancement layers for instance can be be dropped. This will lead to a reduction of the bandwidth usage and the visual quality but allows the uninterrupted decoding and display of the TV channel, while without the use of a scalable codec playback could be interrupted.

Recent developments regarding layered video coding include the development of a layered video encoding profile for the H.264 standard, which is called Scalable Video Coding or H.264/SVC. Figure 2.10 shows an example of Scalable Video Coding. In the example the base layer is extended with one enhancement layer, leading to an improvement of the resolution and the frame rate. This concept could for instance be used to provide IPTV channels in Standard Definition format and then offer an enhancement layer which can be used to upscale the TV channel to High Definition format for users that have sufficient bandwidth available.

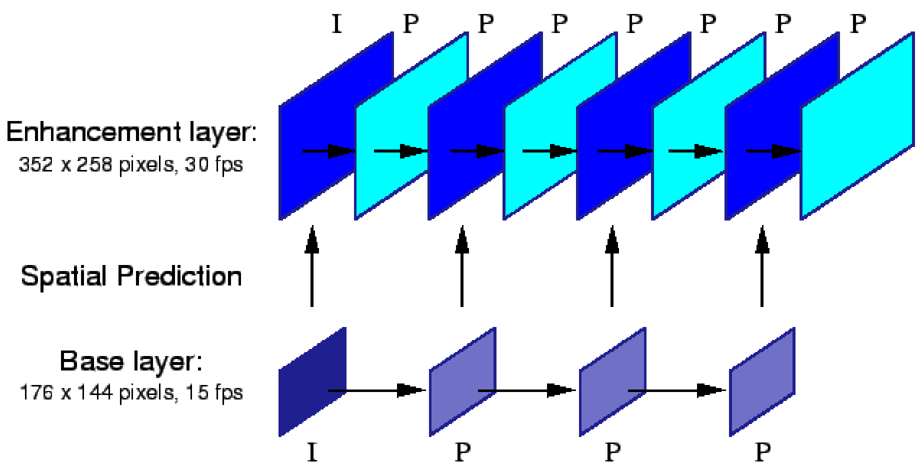

Figure 2.10: Scalable video coding example [45]

2.8 Causes and effects of packet loss

In the previous sections the distribution of IPTV content and the compression of video signals have been discussed. This section focuses on the causes of packet loss and the resulting effects for IPTV applications. Furthermore an impression is given on how packet loss effects the video output that is displayed to the user. The following section then discuss the techniques that can be used to remove or minimize the effects of packet loss.

Packet loss can have different causes:

- Signal degradation over the network medium;
- Congested network links;
- Faulty equipment;
- Faulty routing.

From these four causes, the first two items are the most likely causes for packet loss effecting IPTV applications. The first cause is applicable to DSL and Coaxial cable broadband access networks. A bad quality line leads to degradation of the electrical signal. This can either lead to a signal that cannot be read anymore or erroneous interpretation of the signal. This will cause a packet to get lost either because there is no packet at all, or a packet is dropped because of an incorrect packet checksum. For optical based access network signal degradation is very unlikely and due to protection mechanisms in the link layer data corruption is less likely to occur.

Network congestion is caused when the data throughput approaches the maximum throughput of a network link like an access link or if the throughput approaches the processing rate of a network device like a router. When this device cannot process incoming requests anymore and the packet cannot be buffered it will be dropped. Solutions for congestion include: increasing the network or processing capacity; adaption of the transfer speed or using a QoS scheduling mechanism to prioritize packets containing important data.

2.8.1 The effects of packet loss for IPTV video

When packets of an IPTV stream are lost the decoder in the IPTV set top may not be able to decode the video stream correctly, which leads to visual errors in the displayed video signal. Figure 2.11 shows two examples of the effects of packet loss for streaming video. The image on the left shows how missing data leads to the incorrect placement of parts of the decoded image, as can be seen by parts of the tie and the suit which are relocated to the right. The image on the right shows how missing data leads to strong impairments of the decoded video frame: the head and body of the displayed person are corrupted.

When the payload of a missing packet contains data belonging to a reference frame, corruption of all subsequent predictive frames will occur. They cannot be decoded correctly, or in worst case cannot be decoded at all. The visual impairments will not stop until the next reference frame is received successfully.

Figure 2.11: Examples of visual impairments due to packet loss

This error propagation may continue for multiple seconds, depending on the specified reference frame interval. This issue can therefore have a big influence on the video quality and the Quality of Experience as perceived by the IPTV user. In figure 2.12 the effects of an impaired reference frame and the subsequent propagation of errors is shown. In this example the impairment of one reference frame leads to corruption of in total 19 frames.

2.9 Error resiliency and error correction techniques

As what can be seen from the previous examples, it is important to either prevent packet loss or, when packet loss does happen, restore the missing packets or reduce the noticeable effects of packet loss. In other words, one wants to have a error resiliency mechanism that reduces errors due to packet loss or provide a mechanism to recover from packet loss.

There are numerous ways to provide error resiliency against packet loss. The techniques can be divided in two categories: the techniques that provide recovery for packet loss and the techniques that try to reduce ('conceal') the impact or effect of packet loss. Two common error recovery techniques are forward error correction and packet retransmission, well known techniques to reduce the loss rates or effects of packet loss are packet interleaving, error concealment, prioritization of the application payload and bandwidth adaptation. Error recovery and error concealment techniques can also be combined to further minimize the effects of packet loss.

2.9.1 Forward error correction

Forward error correction (FEC) is the technique of adding redundancy to the data that needs to be transmitted. This redundant data allows receivers to reconstruct the data that is missing. The amount of data that can be reconstructed depends on the amount of redundant data and the amount of loss. A FEC-based recovery mechanism does not require any feedback from the sender to the receiver and is therefore suitable in networks that only allow uni-directional traffic (e.g. satellite) or environments where the latency

(a) Frame 1 (b) Frame 2 (c) Frame 3

(d) Frame 4 (e) Frame 6 (f) Frame 8

(g) Frame 12 (h) Frame 16 (i) Frame 20

Figure 2.12: Error propagation in subsequent video frames. The video corruption starts in frame 2 and lasts until frame 20.

from receiver to sender is relatively high (e.g. cellular networks).

Forward Error Correction can be applied on different levels of the OSI reference model[2], from the physical layer up to the application layer. For IPTV streaming applications, FEC on can be offered as network layer, transport layer or application layer FEC. A widely used FEC scheme is Raptor FEC encoding [46].

The bandwidth overhead needed for the inserted FEC data can be calculated in advance, which means that network operators can take this overhead into account when making bandwidth reservations for IPTV stream delivery. Because additional data is inserted that might be used for the recovery of packets, a FEC scheme introduces some additional delay, but the amount of delay is less then the delay introduced by a

[2]The Open System Interconnection reference model is a framework for designing network protocols. It was defined by the International Organization for Standardization. The framework consists of 7 abstract layers. From top to bottom: Application, Presentation, Session, Transport, Network, Data Link, and Physical layer. Each layer enhances the communication services of the layer directly below and is also shielded from the implementation details of the lower layer(s). More information can be found at: http://www.itu.int/rec/T-REC-X.200/en.

retransmission mechanism.

A disadvantage of a FEC protection mechanism is that the applied FEC protection scheme might be insufficient (too weak) for certain users, while at the same time be superfluous for some users, thus being a waste of bandwidth. In a multicast IPTV distribution network with thousands of concurrent users, this can become an issue.

2.9.2 Adaptive forward error correction

Adaptive forward error correction is a special type of application of a FEC mechanism. The amount of redundant data that is transmitted with the data is adapted to the loss characteristics reported by the receiver(s), but may lead to more efficient usage of the network as the redundant data only transmitted when required. This means that adaptive forward error correction is only applicable in distribution networks that allow giving feedback from receiver to sender. Applying adaptive forward error correction for video streaming applications has been investigated in [47] and [48]. The former focusses on real-time delivery scenarios, while the latter investigates how adaptive FEC can be provided with RTSP and RTP based streaming.

2.9.3 Packet retransmission

Packet retransmission is the technique of retransmitting packets that are considered lost. A packet retransmission requires communication between the receiver of the data and the sender of the data, because the receiver needs to implicitly or explicitly ask the sender for packet retransmissions. A well known protocol using packet retransmission is the Transmission Control Protocol (TCP) [10]. A packet retransmission mechanism requires bi-directional communication to allow the receiver to indicate packet loss and requires the means to identify which packet needs to be retransmitted. This is typically done by applying sequence numbers. To indicate the loss of packets two types of messages can be used:

- NACK or negative acknowledgment messages are used to explicitly indicate that one or more packets where not received. NACK messages are used in networks where feedback from receiver to sender should be kept to a minimum, due to networks constraints or due to a network topology with many receivers per sender.
- ACK or acknowledgment messages can be used to implicitly indicate the loss of a packet by acknowledging the the reception of one packet or a sequence of packets. These messages can then be used to implicitly determine packet loss, as the packet that is missing will never be acknowledged. Implicitly indication of packet loss is for instance used by TCP, which repeatedly sends ACK messages for the highest in sequence received packet.

A packet retransmission mechanism is adaptive to variable network conditions. When there is no loss in the network, there will be no packet retransmission: only when losses occur will the retransmission mechanism require bandwidth. Because packet retransmission mechanisms introduce delay, they are only suitable for applications with non strict delay requirements. The in section 2.2.1 discussed protocols already showed that the suitability of packet retransmissions is based on application requirements, which either prefer reliable delivery or timely delivery, as for some applications like live streaming or telephony the data that is being retransmitted is only valuable for a short amount of time.

A possible drawback of using packet retransmissions for (live) IPTV services is that IPTV streams are not likely to get adapted to congestion. When congestions occurs during transport and packet retransmission is enabled, the packet retransmissions may contribute to the network congestions, thus lowering the available bandwidth for the IPTV stream delivery.

2.9.4 Payload interleaving

Besides the above mentioned techniques for the recovery of missing data, there are other techniques that can be used to minimize the (noticeable) effects for packet loss. For applications like telephony and video streaming the loss of very small amounts of data may not be that problematic, as the data is only valuable for a short amount of time and the loss of small portions of data might not be noticeable for the user. These applications are called loss-tolerant applications. For other applications, like file transfers or on-line banking however packet loss can result in serious problems for the user or service provider.

One approach to minimize the (noticeable) effects of packet loss, is to interleave the data that is being transmitted. By interleaving the transmitted data, the packet loss will only effect small subsequent parts of the transmitted data, therefore minimizing the 'instantaneous' severeness of packet loss in the application output and increasing the possibilities for concealment of errors. Due to the temporal separation of adjacent video frames, the losses are dispersed over multiple frames, allowing for multiple minor errors in multiple frames, that might be imperceptible by the user, instead of one major error that is easily perceived by the user. This can be seen in figure 2.13, which demonstrates how errors are dispersed over multiple frames when payload interleaving is used. Another advantage of the usage of payload interleaving is that smaller losses resulting from interleaving are easier to conceal than losses occurring in one burst.

The performance of an error loss concealment scheme varies inversely with the length of the loss period, so when large burst losses occur packet interleaving may not provide any improvements. Due to the ordering and reordering of interleaved frames additional delay is introduces and larger sender and receiver buffers are required.

2.9.5 Error concealment

Video and audio formats can provide error concealment functionality. The decoder detects loss when decoding the data and conceals the missing or corrupted data by means of specific algorithms. An example for concealment of video data is using data from neighboring regions of an image to "fill" the missing

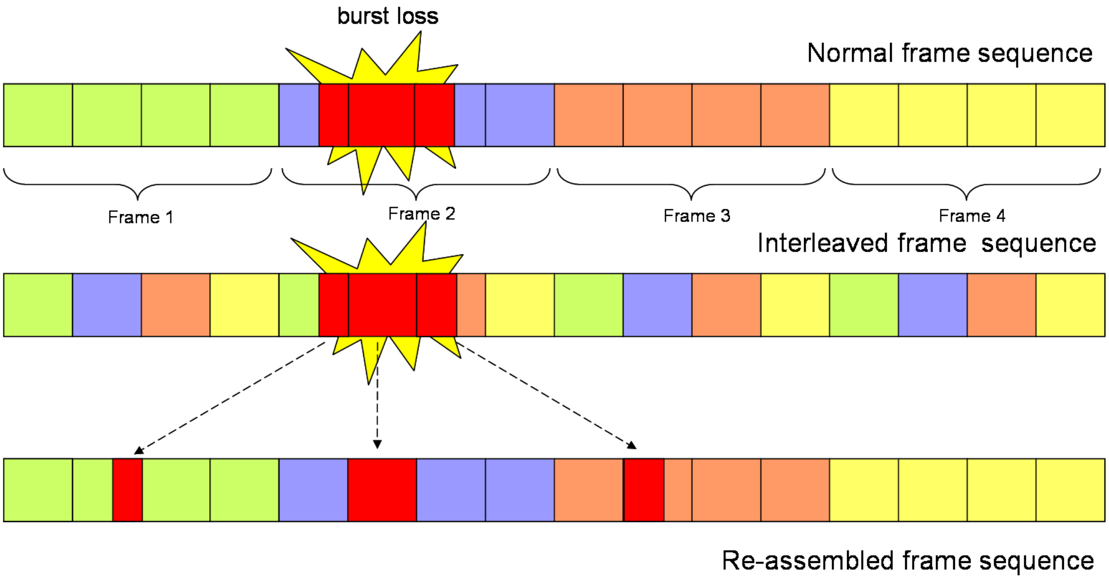

Figure 2.13: Payload Interleaving: due to interleaving the burst losses are dispersed over multiple frames. In a normal situation frame 1 is lost entirely; with interleaving the losses are spread over frames 1 to 3.

data, or use parts from previous or subsequent video frames, as they might contain data that resembles the missing data. Both the MPEG-2 and H.264 codecs contain error concealment functions.

2.9.6 Prioritization of IPTV data

To reduce the effects of packet loss, a service may prioritize some data over less important data. In this section three examples of error reduction based on prioritization are given.

- Video layer prioritization: When a IPTV stream is encoded using a layered video codec such as SVC (see section 2.7.4), a router might decide to drop packets containing data of the enhancement layers, to prevent congestion. The IPTV set top box will still receive the base layer and thus can still decode and display the IPTV channel, with a reduced quality, but the IPTV service remains viewable. A RTP payload format for SVC is currently in the process of being standardized [49].
- Video frame type prioritization: Besides providing different layers of video, other approaches toward splitting the content based on priority exists. For instance the application data units could be prioritized based on importance. For instance, when the IPTV stream consists of MPEG-2 video, the packets can be prioritized based on the frame-type: I-frames can for instance get a higher priority then B-frames.

2.9.7 Bandwidth adaptation

A last form of error resiliency is the adaption to network conditions by reducing the bandwidth used for the IPTV stream or changing the video encoding profile, making the transmission more robust or reduce the network load in congested networks. This might lead to a reduction in the audial and visual quality, but a satisfactory End-to-End service may still be possible. For a multicast distribution network this approach is however not desirable, as the adaption concerns all users, not only one. This mechanism is however not suitable for multicast streaming networks, as the adaption will impact all multicast receivers, something which is not desirable.

2.10 Quality measurement and management

For real-time services like IPTV broadcast television the quality of the provided service is an important aspect. When the provided service quality does not satisfy the customer, the customer will not use the service. The minimum requirements for a specific service are typically defined as a Quality of Service or QoS level. QoS is often defined and enforced by network-based service level agreements by means of objectively measured network metrics like delay, jitter, latency and packet loss. An IPTV service may for instance have a maximum acceptable startup delay requirement or a recommended acceptable packet loss rate.

QoS metrics often focus on the objective quality as perceived and measured by the service provider. These metrics however do not neceserraly correspond with how a certain service is being perceived by the end user: there is not a simple relation between the quality of the network and the quality of the video as perceived by the end user. This subjective user experience is becoming more and more important, especially in the case of time constrained multimedia applications like IPTV, where responsiveness and visual quality may be key factors for users to like or dislike the offered service. Especially when users have a certain expectancy of the service, which is the case for IPTV, as users are accustomed to the quality provided by 'traditional' analogue cable broadcast TV. This subjective measure for the user experience is often addressed as the Quality of Experience or QoE. QoE measurements and guidelines for IPTV services have been specified by several standardization bodies, for example the DSL forum [50] and the ITU-T [51].

When looking specifically at QoS and QoE metrics for IPTV services, the following categories can be distinguished:

- metrics related to the transport and delivery of the IPTV service (network quality metrics);
- metrics related to the video and audio quality of the IPTV service (video and audio quality metrics).

The next section will focus on network quality metrics and recommended network requirements. In section 2.12 the video quality metrics will be discussed and different types of video quality measurements will be explained as well.

2.11 Network quality metrics

For describing the network quality the following metrics are generally used: bandwidth, latency or delay, jitter and loss. A description of the metrics is presented in table 2.3. These metrics generally determine if a network is suitable to transport data for a certain type of application. For instance, for real-time applications, like VoIP, low delay and jitter values are very important, as when these values get too large, a VoIP service is not possible anymore. For bandwidth-sensitive applications like (real-time) IPTV video streams, the available bandwidth is important as when the required bandwidth cannot be offered, the IPTV service cannot operate correctly. For elastic applications like file transfers, web browsing and e-mail, the available bandwidth is not a big concern; they can operate with as little as bandwidth available (it only takes more time to transfer the data, which leads to larger waiting times for the end-user).

Metric	Description
Bandwidth	The amount of data that can be sent over a network connection in a given period of time. Bandwidth is usually measured in bits per second
Latency	The time it takes for a data packet to go from the sender to the receiver. Latency is also addressed as delay. Two types of latency measures are common: single-trip delay specifies the one-way delay from sender to receiver while the round-trip time delay specifies the one-way delay from sender to receiver plus the one-way delay from receiver to sender
Jitter	The variability over time of the latency across a network. The short term variation of a digital signal's significant instant from their ideal positions in time
Loss	The measure of the number of packets lost in the communication between sender and receiver. Loss is often addressed as a percentage of the total number of packets transmitted or the number of packets lost in a specified time interval

Table 2.3: Network metrics

The in table 2.3 mentioned metrics and metrics deducted from these metrics have been standardized by the IP Performance Metrics (IPPM) Working Group of the Internet Engineering Task Force (IETF) [52], whom's goal it is to develop a set of standard metrics that can be applied to the quality, performance, and reliability of Internet data delivery services. The IPPM Working Group currently has developed metrics for:

- one-way and round trip delay (RFC 2679 and RFC 2681)
- one-way packet loss (RFC 2680)
- delay variation (jiter) (RFC 3393)
- loss patterns (RFC 3357)
- packet reordering (RFC 4737)
- link bandwidth capacity (RFC 5136)

2.11.1 Network quality requirements for IPTV services

High quality video services generally have strict requirements regarding packet loss, jitter and delay. For IPTV services packet loss rates of 10^{-4} to 10^{-4} or less, latency in the order of hundreds of milliseconds and jitter on the order of a few tens of milliseconds may be tolerated [53]. Exact requirements for video services have not yet been established, but there are several guidelines available, amongst others from the ITU-T and the DSL Forum IPTV bodies. Two examples are given below. The ITU-T developed an informative classification of the packet loss rate for digital video services (see table 2.4). This can be used to determine the service quality that can be achieved based on the available loss rates in the network, or determine the network requirements needed to provide an excellent service quality.

Packet Loss Rate	QoS
$PLR \leq 10^{-5}$	excellent service quality (ESQ)
$10^{-5} < PLR \leq 2 * 10^{-4}$	intermediate service quality (ISQ)
$2 * 10^{-4} < PLR < PLR\_out = 0.01$	poor service quality (PSQ)
PRL_out = 0.01 < PLR	IP end-to-end service not available

Table 2.4: Informative classification used for digital television services, from ITU-T J.241 Appendix A [54]

In technical report TR-126 of the DSL Forum QoE requirements for triple play services are provided, which include IPTV requirements and recommendations. Table 2.5 shows recommendations for transport layer parameters for various MPEG-2 bit stream formats for Standard Definition television. The document also gives recommendations for other video formats and High Definition video.

Transport stream bit rate (Mbps)	Latency	Jitter	Maximum duration of a single error	Corresponding Loss Period in IP packets	Loss Distance	Corresponding Average IP Video Stream Packet Loss Rate
3.0	<200 ms	<50 ms	<= 16 ms	6 IP packets	1 error event per hour	<= 5.85^{-06}
3.75	<200 ms	<50 ms	<= 16 ms	7 IP packets	1 error event per hour	<= 5.46^{-06}
5.0	<200 ms	<50 ms	<= 16 ms	9 IP packets	1 error event per hour	<= 5.26^{-06}

Table 2.5: Recommended minimum transport layer parameters for satisfactory QoE for MPEG-2 encoded SDTV services. From: DSL Forum Technical Report 126 [50]

2.12 Video quality metrics

Packet loss can lead to visual impairments, but there is no way to easily determine the effect of packet loss on the resulting video quality, or the perceived quality, as the effect of packet loss depends on multiple factors:

- The video codec;
- The codec settings, including the the Group of Pictures (GOP) size. The GOP size determines the amount of frames that have a temporal relation with the first frame in the GOP. The bigger the GOP, the higher the propagation of visual impairments.
- The type of frame the lost data belongs to. For MPEG-2 encoded video, the loss of the data belonging to a B-frame will only lead to impairments in one frame while loss of data belonging to a P-frame can lead to impairments to previous or subsequent P or B-frames in the same GOP. The loss of data belonging to a I-frame will lead to visual impairments to all subsequent frames up the start of the next GOP.
- The compression ratio;
- Error concealment functionality of the video codec. When concealment is used users may not notice visual impairments when packet loss occurs.

Similar to the network metrics, standardization of video metrics is taking place, such that unambiguous video metrics are defined and that measures based on the metrics can be exchanged by interoperable products. Two examples of standardization are the IPTV QoS/QoE Metrics being created by the IETF working group on IPTV and the creation of an RTP extension to provide video metrics using extended

RTCP reports.

IPTV QoS/QoE metrics

The proposed IPTV QoS/QoE metrics by the IETF working group on IPTV cover several topics [55]. They are categorized as follows:

- **Perceptual Quality Metrics** provide high level video and audio QoE scores, giving visibility of the impact of a wide range of impairments.
- **Video Stream Description Metrics** provide information on the type of video codec being used, Group of Pictures structure and length, image size and other key factors
- **Video Stream Metrics** provide insight into the proportion of different types of video frames that are impacted by packet loss.
- **Transport Metrics** provide insight into essential data regarding jitter, delay and packet loss similar to the network metrics described in section 2.11. The metrics also include information about the effectiveness of error correction mechanisms such as FEC or packet retransmissions.
- **MPEG Metrics** provide information about the MPEG transport stream (MPEG-TS) being used for transport of the video content. Metrics related to MPEG transport stream parameters are provided as well.

RTCP extended reports

In 2003 the Extended Report (XR) packet type for the RTCP protocol was defined [56]. This extended report can be used to provide information beyond what is possible with the reception blocks available in RTCP sender and receiver reports. Examples are providing detailed information about packet loss, discard and delay metrics. The standard furthermore address Voice over IP (VoIP) related metrics regarding signal, noise and echo levels. Currently a new report block type for this extended report is in development, which allows to provide QoS metrics for video over IP services [57]. The proposed report block provide the IPTV QoS/QoE metrics mentioned above, but also covers playback related metrics, like the number of playback interrupts, delay between audio and video streams, and the playback buffer size. Because it is still being developed the contents and metrics covered by the report block may still change.

2.13 Video quality measurement techniques

There are three different methods to measure video quality for IPTV services:

- Objective measurements - by comparing the video signal that is transmitted over the network with the source video signal.
- Subjective measurements - by using controlled video experiments in which participants rate the video quality by using a predefined scale.

- Indirect measurements - by using network measurements. The video quality is measured based on an estimated impact of network impairments.

2.13.1 Objective measurements

Objective measurements compare the output video signal with the source video signal. The differences are then used to determine how much the decoded stream deviates from the original. The bigger the difference, the lower the quality of the received stream. There are three types of reference based measurements:

- Full reference - For full reference the source video signal is completely (frame by frame) compared with the received and decoded video signal. Both streams need to available for comparison, therefore this type of measurement is not usuable in a deployed IPTV network, as a IPTV client only receives one stream.
- Reduced reference - measurement where partial extracted information from the transmitted signal and the entire received video signal are available for comparison. The extracted information is transmitted with the IPTV stream or provided externally.
- Zero reference - measurement where information from the transmitted signal is not available; only the received signal can be used for measurement. This solution is often used in environments where the source stream is not available, which is typically also the case for IPTV broadcast TV. One example of a zero reference based measurement is based on measuring block edge impairments in decoded video [58].

Reduced and zero reference based measurements are often codec dependent, as they make use of specific codec characteristics to detect impairments.

Peak Signal to Noise Ratio (PSNR)

One full reference object measurement technique that is often used to evaluate video quality is by computing the peak signal to noise ratio or PSNR. The PSNR is computed by taking the root mean square (RMS) value of the differences (errors) of the original and the received video frames, often normalized to be expressed in dB. PSNR measurements are often used to compare the quality loss of a video codec compared to a the raw video footage, but can also be used to compute the quality loss due to packet loss. A PSNR ratio of 34dB and higher is required for television broadcast; lower than 30dB is not acceptable anymore and anything below 20dB can be considered unwatchable [59]. Although PSNR measurements can tell to what extend a video stream has been impaired by for instance packet loss, the measurement do not provide any information about the temporal aspects of a received and decoded video stream: the measurement does not tell anything about frame rate drops, freezes during playback or delayed playback caused by buffer underruns, which will also influence the perceived user experience.

A disadvantage of PSNR based quality measurements is that it is a computation-intensive operation, as it requires the comparison of every decoded video frame with the source video frame. Another constraint is that PSNR requires access to the original video stream and the decoded, possible impaired video stream, which in a live IPTV broadcast scenario is not possible. A different issue is that PSNR do not necessarily have to correlate with the quality as perceived by the end user.

2.13.2 Subjective measurements

Subjective measurements use human viewers to rate quality of a video sequence. Subjective measurements can provide an accurate assessment of the video quality, as it reflects the video experience the end user will experience as well. However, from a practical point of view they are hard to use, as they require interaction with the user. The mostly used subjective quality measurement technique is Mean Opinion Score.

Mean Opinion Score

Mean Opinion Score or MOS is a subjective measurement indication, which ranks the video quality based on user feedback. In MOS measurements users determine the quality by rating the quality of the displayed video sequence on a scale of 1 (very bad) to 5 (very good). The averages of different users are taken and so the MOS value is computed. In comparison with PSNR measurements, subjective measurements do take spatiality (time) into account, as hick ups and video playback freezes will be noticed by the test subjects. In table 2.6 the MOS values with a quality and impairment scale for television broadcasts are presented, as defined by the ITU-R [60]. A possible conversion scale for PSNR values to MOS values [61] is also presented in table 2.6.

MOS	Perceived Quality	Impairment	PSNR (dB)
5	Excellent	Imperceptible	> 37
4	Good	Perceptible, but not annoying	31-37
3	Fair	Slightly annoying	25-30
2	Poor	Annoying	20-24
1	Bad	Very annoying	<20

Table 2.6: ITU-R Quality and Impairment scale with MOS to PSNR conversion

2.13.3 Indirect measurements

The final category of video quality measurements for IPTV services is based on indirect measurements. In this case impairments of network parameters are used to predict the resulting video and service quality for the IPTV service. Currently the ITU-T IPTV focus group is working on specifying Quality of Experience metrics for IPTV [55] and defining how these metrics should be measured. The specification also includes indirect measurements. One of the proposed techniques is using an estimated PSNR (EPSNR) to determine video quality metrics [62]. This estimation uses the relation between the loss of a packet and the proportion of pixels that are impaired due to the packet loss. The model also takes into account the type of frame that is impaired. This techniques looks promising as this technique can be applied in practice, even for live video broadcasts, without requiring user input.

2.14 Summary and conclusions

In this chapter an introduction to IPTV was presented and the many possibilities for delivering IPTV services were discussed. There are several network protocols that can be used for IPTV stream delivery, but the suitability of a specific protocol depends on a number of factors:

- the application scenario;
- delay requirements of the IPTV application;
- the configuration of the network.

Two suitable protocols for multicast IPTV stream delivery are the User Datagram Protocol, a simple protocol which does not provide a feedback mechanism, and the Real-Time Transport protocol, a transport protocol that runs on top of other transport protocols, which provides mechanisms for providing feedback about the reception of data. This mechanism however is not suitable for reporting (time-constraint) feedback for a RTP session with many participants, which is the case for multicast IPTV stream delivery.

The chapter also discussed video compression principles and examples of video formats currently being used for IPTV services. An answer to the the research question: *What are the effects of packet loss on IPTV streaming applications?* can now be given. Packet loss leads to visual impairments of the decoded video stream and due to the principles of video encoding, especially related to the temporal relation between subsequent video frames, errors that operate in reference video frames can propagate in subsequent frames that need the reference frame to decode. A single lost packet can thus result in the corruption of multiple video frames which may leads to a lower Quality of Experience for up to several seconds.

The impact of packet loss on the resulting video quality can be measured in different ways:

- Objectively by comparing the source video signal with the video signal provided to the user.
- Subjectively by letting test persons rate the resulting video quality.
- Indirectly by looking at the network impairments and their expected effect on the video quality.

These techniques can also be used to evaluate the performance of error resiliency mechanisms.

Various error resiliency mechanisms were presented that can be used to reduce the noticeable effects of packet loss for the user. Recovery of missing data can be provided by means of Forward Error Correction or packet retransmissions.

A FEC mechanism allows the IPTV stream receivers to recover a limited amount of missing data. When in a large multicast group the loss rates vary for different users there will either be some users with remaining losses or bandwidth will be wasted in large parts of the network where the loss rates are low.

On the other hand is an end-to-end packet retransmission mechanism for a large multicast distribution network for a time-constrained IPTV service not preferable as feedback implosion leads to performance issues when receivers ask the the source of the IPTV stream for packet retransmissions. Furthermore can the high latencies for end-to-end packet retransmissions make it difficult to provide rapid recovery of missing packets.

Another solution is to provide packet loss recovery for smaller parts of the multicast distribution tree. By introducing a fast-retransmission function in a subtree of a multicast distribution network, feedback implosion can be reduced and loss recovery can be provided rapidly allowing the video quality for the users to be maintained.

This however requires a packet retransmission mechanism adequate for time-constrained multicast IPTV stream delivery, i.e. a mechanism that favors timeliness over reliability and can be used in combination with a transmission protocol suitable for multicast IPTV stream delivery, such as RTP. Although the RTP protocol is not equipped for applying packet retransmissions, two protocol extensions allow RTP to be used to provide time constrained feedback and offer retransmissions with an RTP retransmission packet format. These extensions can be used to design a RTP packet retransmission mechanism for multicast IPTV stream delivery.

Chapter 3

Requirement analysis

In this chapter the requirement for a fast retransmission mechanism for multicast IPTV will be presented.

In section 3.1 the context of the work described in this thesis is provided. Section 3.2 discusses the scenario for a broadcast television IPTV service, which uses a multicast IP service for the delivery of IPTV streams. Section 3.3 states the technical details of the scenario and provides the assumptions on which the requirements are based. Finally, section 3.4 provides the requirements for the prototype and the justification for the requirements.

3.1 Company requirements

The work described in this thesis is part of a IPTV functions study for GPON access network equipment performed by NEC Eurolabs in 2007/2008 [63]. One of the proposed functions is applying error recovery for multicast IPTV services like live television broadcasts by means of packet retransmission. This function was also identified in the IPTV literature study that was performed as preparation to this master thesis [64]. This thesis further investigates this function, by means of the design and implementation of a prototype for packet retransmission for a multicast IPTV service, which allows for an evaluation of the capabilities of the proposed function.

3.2 Scenario description

An IPTV service provider offers linear broadcast TV channels to its subscribers, which are being transmitted via a multicast distribution network, as seen in figure 3.1. The distribution network is managed and controlled by the IPTV service provider, i.e. the services are not offered via the public Internet. All IPTV channel streams are constantly available in the core network of the IPTV service provider and are only sent to the respective IPTV subscriber when the subscriber starts viewing the specific TV channel. When a subscriber selects a TV channel, the Set Top Box will join the multicast group which is used to for the transmission of the IPTV stream. The packets that are sent to the multicast address will via the access link be forwarded to the subscriber's Set Top Box. The Set Top Box will decode the packets and display the television channel on a screen.

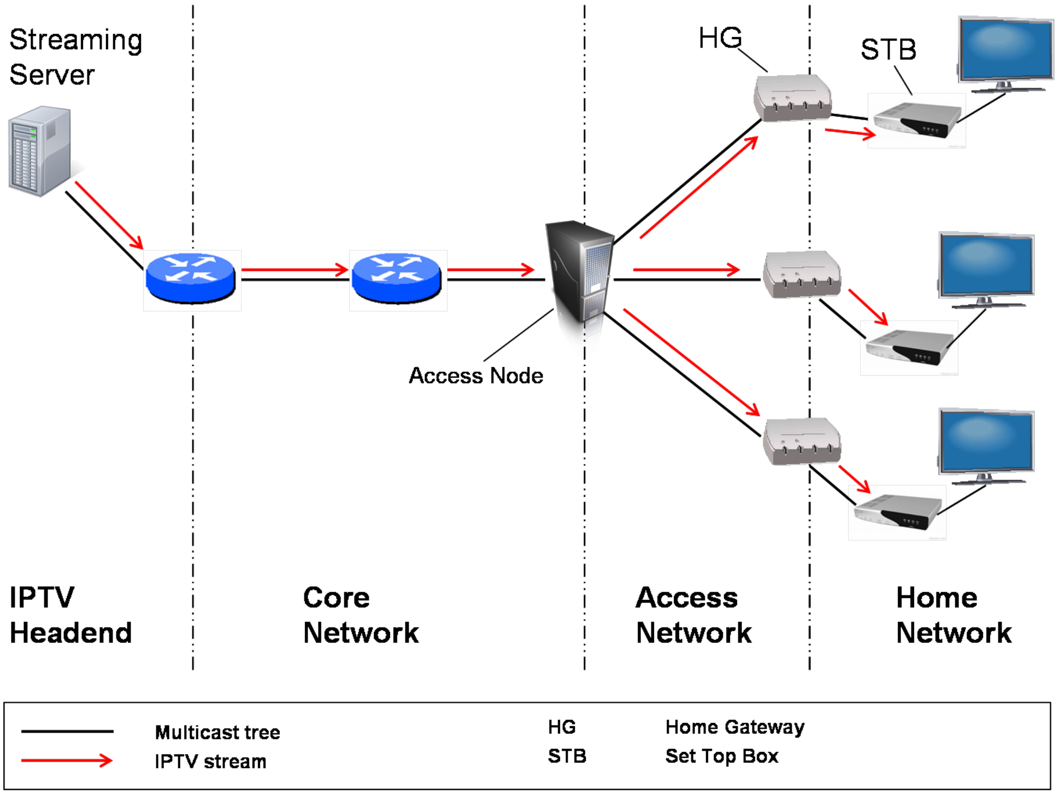

Figure 3.1: IPTV distribution network

IPTV subscribers can have different types of access networks: DSL, coaxial cable and GPON are one of the options. The quality of the different access links from subscriber to the core network may differ and some access links may suffer from packet loss. This packet loss can have a severe influence to the IPTV service, manifested in visual impairments or playback problems, leading to an unsatisfactory TV viewing experience.

To resolve packet loss packet originating in the access network packet retransmission may be offered, such that the QoS of the IPTV service can be maintained. This can either be due to full packet recovery or due to a significant reduction of the packet loss rate, allowing the ratio to stay above a defined acceptance threshold. One approach for providing packet recovery is temporarily caching the packets that are forwarded to the subscribers in an access node such as a MSAN or DSLAM, allowing localized resiliency against packet losses occurring between the subscriber and the core network.

When during playback a subscriber's set top box detects packet loss it can immediately asks for retransmission of the missing packet(s), which can be offered by the access node in a very short time span (as the propagation delay only consists of the access network). This allows the Set Top Box to place the retransmitted packet in its reception buffer, before the data needs to be processed for displaying.

An example of this described retransmission functionality is shown in Figure 3.2. In this example the access node provides temporary caching of packets which are destined for the IPTV subscriber. The IPTV clients can request packet retransmissions for packets that are lost in the access network or home network.

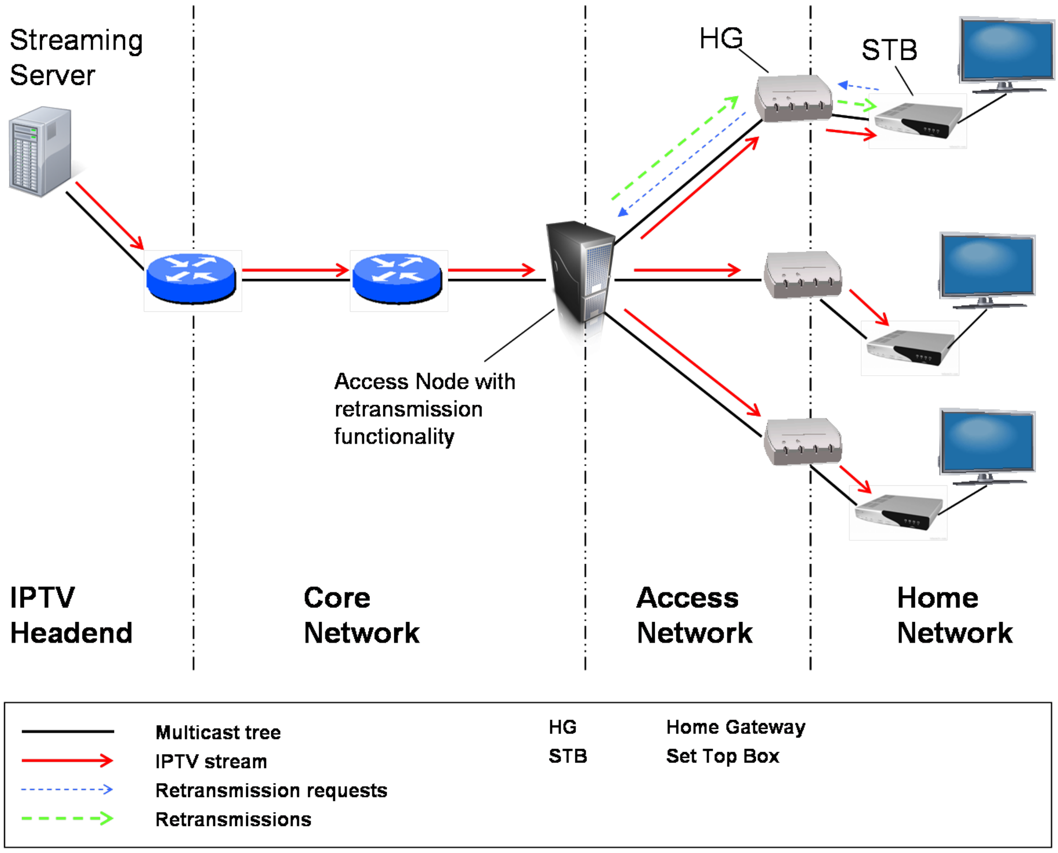

Figure 3.2: IPTV distribution network with retransmission functionality provided by the Access Node

In this scenario the IPTV service provides live broadcast television. This means that the timespan to recover a packet is strictly bounded by the amount of delay that may be introduced for the television service before playback can start.

3.3 Technical description

The retransmission functionality will be provided on a specific subtree of a multicast IPTV distribution tree. For the distribution of a multicast IPTV channel, the protocol stack presented in figure 3.3 is assumed. This assumption is made on the information described in Chapter 2, the IPTV literature study that preceded this thesis [64] and information provided by NEC.

The figure shows the end-to-end delivery of the IPTV content from the streaming server to the Set Top Box. The intermediate components only provide functionality up to the network layer, in this case IP: they provide the routing and forwarding of the IP packets. The IPTV payload, consisting of audio and video streams and optionally other multimedia data, are transmitted using a connectionless, multicast capable transport protocol. Note that in this stack the transport layer is not yet defined, as there are several transport protocol possibilities. These have been discussed in Chapter 2.

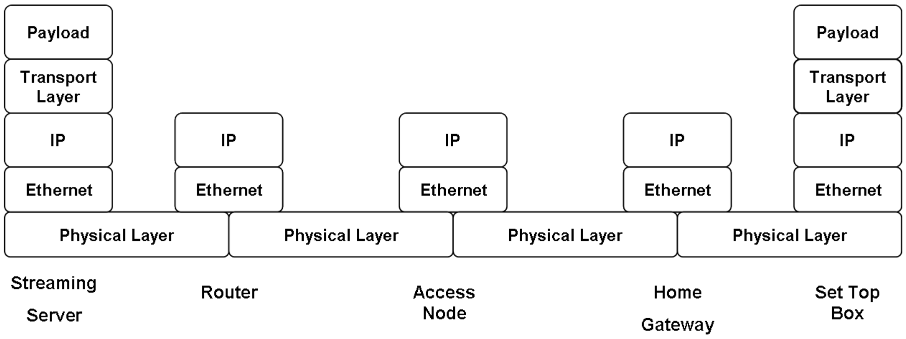

Figure 3.3: IPTV distribution protocol stack

3.3.1 Assumptions

The following assumptions are used for the requirements presented in section 3.4.

- The access network links to the end user provide sufficient bandwidth for the transmission of a IPTV broadcast channel. Otherwise IPTV services cannot be offered to the user at all.
- The packet loss occurs between the retransmission cache and the IPTV client. The proposed solution will not offer resiliency for packet loss occurring between the streaming server and the retransmission cache. It is assumed that this part of the distribution path is part of a managed network where bandwidth reservations for the distribution of the IPTV channels has been made, or the core network contains redundancy, to reduce the probability of packet loss effecting the IPTV users.
- A IPTV multicast distribution network as described earlier is assumed. Besides the Streaming Server, Retransmission Cache and IPTV client, this distribution network consists of multicast routers which are capable of forwarding the data from a multicast channel to a Set Top Box.

3.4 Requirements

Based on the company requirements, the scenario description, the technical description and the presented assumptions the following requirements are formulated:

1. The system shall provide multicast IPTV streaming with error resiliency based on packet retransmission. The component that offers retransmission of packets, the retransmission cache, will be placed in a access node of the multicast distribution network.
2. The IPTV multicast stream will be sent by a Streaming Server and decoded and displayed by an IPTV Client.
3. The retransmission mechanism shall provide packet retransmission as an addition to the 'normal' IPTV stream delivery, using the distribution network described in section 3.2.
4. The packet retransmission mechanism shall provide error resiliency to packet loss originating in the access network.
5. To allow for packet retransmissions the IPTV packets need to have support of sequence numbers.

3.4.1 Subrequirements

The system thus requires three components: A Streaming Server, a Retransmission Cache and a IPTV client. In the following subsections the subrequirements for each of the components are defined:

Streaming server

1. The IPTV Streaming server offers multicast based streaming of IPTV channels.
2. The transport protocol used by the IPTV Streaming server supports packet identification, by means of sequence numbering.

Retransmission Cache

1. The Retransmission Cache must support the reception of IPTV packets that are sent via multicast.
2. The Retransmission cache must be able to temporarily buffer incoming packets to allow these packets to be available for retransmission.
3. The Retransmission cache must be able to receive and interpret retransmission request sent by IPTV clients.
4. The Retransmission cache must be able to transmit retransmission packets to a IPTV client upon requests made by a IPTV client using a retransmission request.

IPTV client

1. The IPTV client must support the reception of IPTV streams which are transmitted via multicast.

2. The IPTV client must be able to detect packet loss.
3. The IPTV client must be able to ask for retransmission of packets that are considered lost.
4. The IPTV client must be able to receive retransmission packets and place these packets in order in the packet buffer.
5. (optional) The ITPV client must be able to determine if the retransmission of a packet can be performed before the respective packet is needed by the application.
6. The IPTV client must be able to decode and display multicast IPTV, where error recovery by means of packet retransmission may be applied.

3.5 Justification

The rationale for specifying these requirements are the following:

- Requirement 1 is based on the function study performed by NEC.
- Requirement 2 states the IPTV components used in a IPTV distribution network, as described in Chapter 2.
- Requirement 3 is chosen to specify a packet retransmission scheme that can easily be introduced in an existing IPTV multicast distribution network. In other words, the retransmission scheme does not involve the source of the IPTV stream (i.e. the streaming server). This thus leads to greater flexibility of the retransmission mechanism.
- Requirement 4 addresses another aspect of the functions study and the background literature: packet loss is likely to occur in the access network, due to the various different access link type and access network, while the core network offers a higher Quality of Service.
- Requirement 5 is an implicit requirement of retransmission functionality, which as already discussed in Chapter 2: to be able to provide retransmissions, packet identification is required.

The subrequirements for the components are based on the system requirements; they state what requirements the specific components must fulfill such that the overall requirements are fulfilled.

Chapter 4

Prototype design and implementation

In this chapter the design and implementation of a prototype for a fast retransmission mechanism for multicast IPTV broadcasts are given. Section 4.1 discusses the design of the prototype, including the component decomposition. In section 4.2 the retransmission protocol is presented. This section also provides the protocol messages and the algorithms required to provide packet retransmission for the multicast IPTV service. Section 4.3 focuses on the implementation of the design in a prototype setup.

Justification
The primary goal of designing and implementing this protocol is to provide a *proof of concept*, i.e. to show if packet retransmission for multicast IPTV television broadcasts can be applied successfully. Secondary, the prototype can be used to determine the minimum (application) constraints that need to be satisfied before packet retransmission can be successfully applied. These constraints include:

- The IPTV client packet buffer size;
- The Retransmission Cache buffer size;
- The maximum round trip time between IPTV client and Retransmission Cache.

Finally, the effectiveness of packet retransmission mechanism for the IPTV service can be measured under different network circumstances, which can be used to conclude which parameters influence the performance of the retransmission mechanism.

4.1 Design

4.1.1 System composition

In chapter 2 two protocols suitable for multicast IPTV streaming were discussed: UDP and RTP. As the retransmission mechanism requires the identification of missing packets, using only raw UDP for IPTV stream delivery is not sufficient; alternative an additional header could be used to provide sequence numbers, but this functionality is already provided by RTP. Furthermore does RTP provides a feedback mechanism, by means of the RTCP protocol. Therefore the IPTV streams will be transmitted using RTP on top of UDP and the RTCP protocol will be used to report packet loss.

Chapter 2 also discussed the MPEG-TS container format. This will be used to multiplex the audio and video streams of the IPTV content, allowing for the transmission of only one RTP stream containing the audio and video streams. Therefore synchronization of separate RTP streams is not required, reducing the complexity of the system components, but also of the to be introduced retransmission mechanism.

These design choices lead to an update of the protocol stack presented in figure 3.3 to support RTP/RTCP and MPEG-TS. The resulting protocol stack is presented in figure 4.1.

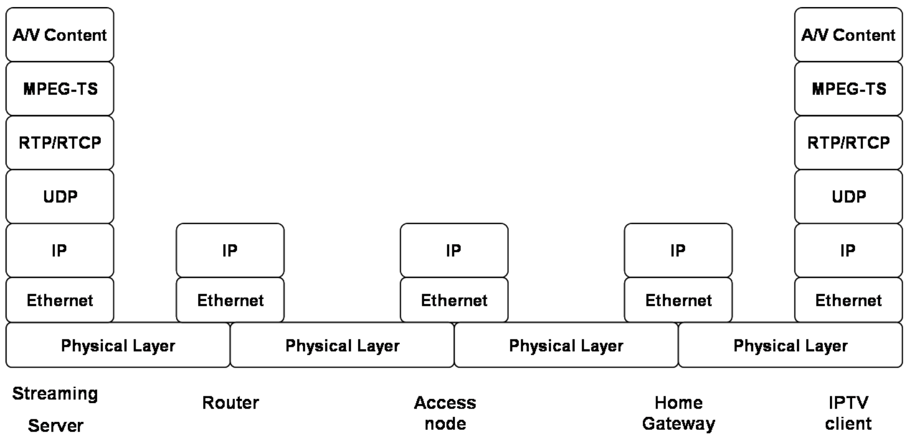

Figure 4.1: Multicast IPTV distribution protocol stack with RTP and MPEG-TS

To allow the access node to buffer packets for retransmission, this access node needs support for UDP, RTP and RTCP as well, as shown in figure 4.2.

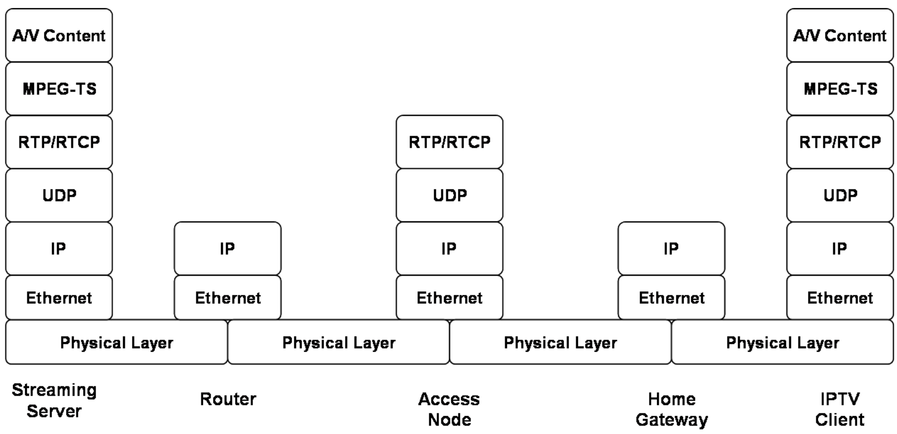

Figure 4.2: Multicast IPTV distribution protocol stack with packet retransmission

What needs to be clarified is how the Retransmission Cache will receive the packets from the IPTV stream. The packets that are sent by the Streaming Server are received by the IPTV client. The Retransmission Cache receives the packets because the packets are processed by the access node. This allows the Retransmission Cache to transparently 'snoop' the packets and store them in a cache. They are then are kept temporarily available for retransmission. When the IPTV client notices the loss of a certain packet, it can ask the Retransmission Cache for retransmission. This is achieved by sending a RTCP report indicating the missing packet(s).

RTCP transmission modes

There are two approaches to transmit the RTCP reports containing retransmission requests: the reports can be transmitted normally, i.e. not directed to the Retransmission Cache, or they can be transmitted to the Retransmission Cache directly. For the first approach the retransmission functionality is provided transparently: the RTCP packets are transmitted to the same address that would be used for normal RTCP receiver reports. This can for instance be the RTP sender (the Streaming Server) or a RTCP aggregation node, which aggregates reports from multiple IPTV clients, for example using one of the approaches described in section [33]. When the RTCP reports are transmitted, the Retransmission Cache snoops the packets when they are processed by the Access Node. This allows the Retransmission Cache to operate transparently.

For the second approach the RTCP reports are explicitly sent to the Retransmission Cache directly, i.e. the IPTV clients need to know the network address of the Retransmission Cache to ask for retransmissions. This can also be combined with the aggregation of RTCP reports.

RTP retransmission modes

In the discussion of the RTP retransmission payload format in section 2.5.3 the two possibilities for transmitting retransmission packets were mentioned. For the prototype session multiplexing will be used. The retransmission packets are sent to the IPTV client(s) using a different network address (consisting of a different IP address and/or port number) as the packets originating from the IPTV Streaming Server. In total thus two RTP sessions will be used; one for the transmission from Streaming Server to IPTV client (and Retransmission Cache) and one from the Retransmission Cache to the IPTV client. An IPTV client is able to relate the two RTP sessions as for the RTP packets of the IPTV as well as the RTP retransmission packets the same synchronization source (SSRC) identifier is used (see also section 4.2).

Because the retransmission are provided in a different RTP session, support for the RTP retransmissions can be optional; only clients capable of receiving retransmitted packets and which are interested in receiving this packets, will need to join a retransmission session.

The prototype composition with the data streams and the different networks is presented in figure 4.3.

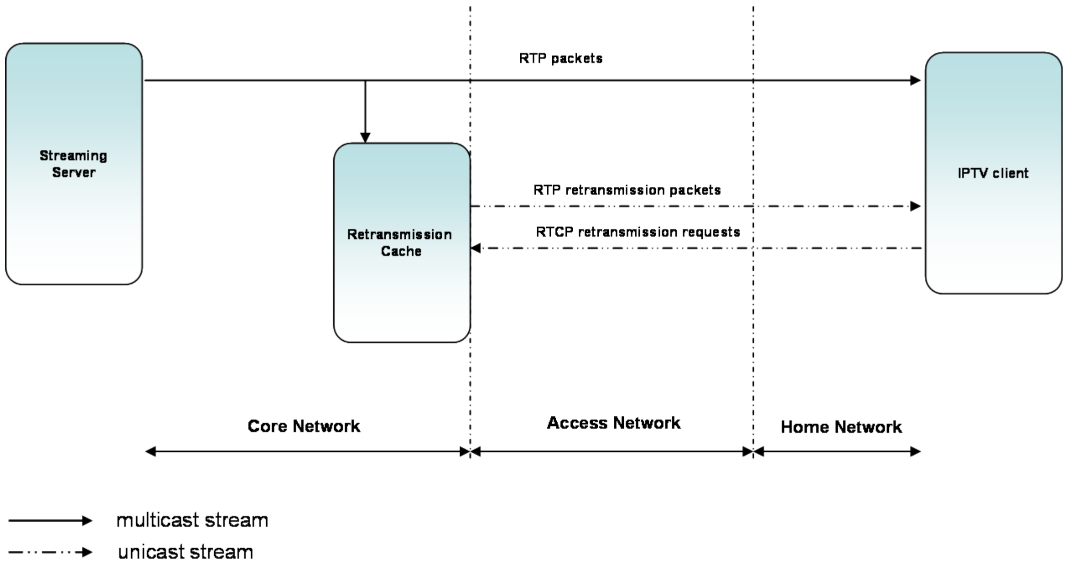

Figure 4.3: IPTV Retransmission Components

4.1.2 System decomposition

Figure 4.4 shows a decomposition of the prototype system and shows the interaction between the different subcomponents.

The Streaming Server consists of three components: the multimedia processing module is responsible for packetizing the audio and video contents in MPEG-TS packets and storing these packets in the sender buffer. The RTP transmitter reads from the buffer and sends the MPEG-TS packets to the IPTV client using RTP.

The IPTV client consists of six components. There are two RTP receiver sockets: one for the IPTV stream packets and one for the retransmission packets. The retransmission logic components is responsible for the detection of packet loss and for generating retransmission requests. These are send to the Retransmission Cache by the RTCP transmitter. The packets that are received, either by normal transmission or packet retransmission, are placed in the receiver buffer. This receiver buffer is being read by the multimedia processing subcomponent, which decodes the audio and video content and outputs it to speaker and display.

The Retransmission Cache consists of five components. The RTP receiver receives the packets being send by the Streaming Server and puts the packets in the Retransmission buffer. The RTCP receiver listens for incoming RTCP packets and forwards retransmission requests to the retransmission logic. This component checks if the requested packets are still available in the buffer and sends the available packets using the RTP transmitter.

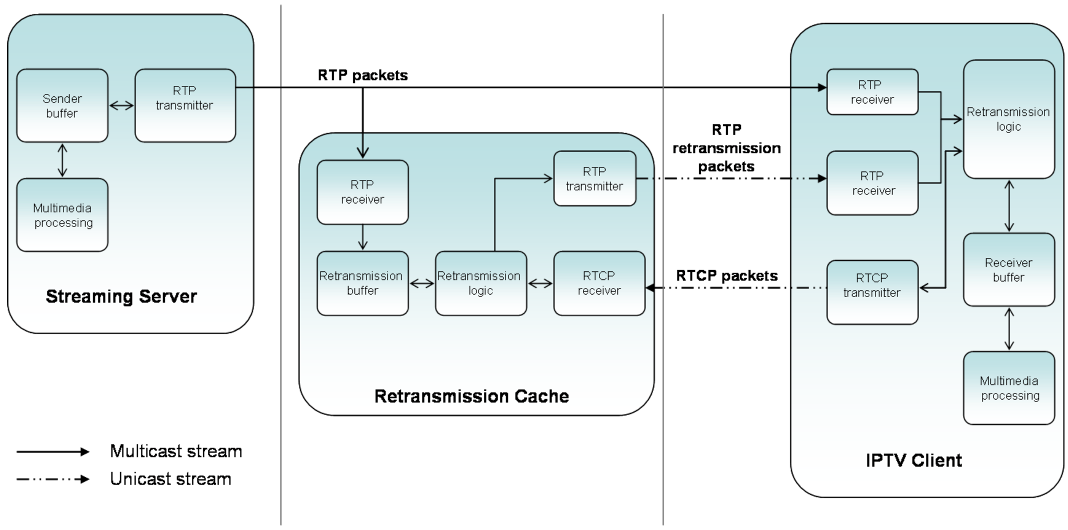

Figure 4.4: Prototype Decomposition

4.2 Retransmission protocol

The IPTV client retrieves packets originating from the IPTV streaming server. It has a packet buffer to store a limited number of packets, to compensate for jitter and, because audio and video frames can only be processed when arrived in total, allow entire application frames to be available for processing by the application. An additional advantage of a packet buffer is that it can be used for packet retransmission.

When the IPTV client identifies the loss of one or more packets, it will create one or more retransmission request message and transmits it to the Retransmission Cache.

The Retransmission Cache continuously receives the packets originating from the Streaming Server and stores the packets temporarily in a cache. The Retransmission Cache also continuously monitors for incoming retransmssion request messages. When a retransmission request message is received, the Retransmission Cache will check if the packets identified in the message are in the cache. If so, the Retransmission Cache will transmit the packet to the IPTV client. Upon reception of a retransmission packet, the IPTV client will put the packet in the right position in the buffer or drop the packet when it is received too late.

4.2.1 Retransmission protocol messages

For packet retransmission two message types are required:

- A retransmission request, used to ask for the retransmission of one or more packets.
- A retransmission response, used to retransmit a packet to an IPTV client.

Retransmission request

A retransmission request is transmitted in a RTCP message. The retransmission request message contains the sequence numbers of the packets considered lost. For this retransmission request message the feedback message type as defined in RFC 4585 will be used. As discussed in section 2.5.2, the RFC specifies a common RTCP packet format for feedback messages, including two payload formats for specifying the type of feedback: Transport Layer (205) or Payload Specific (206) respectively. Additionally there is a variable size Feedback Control Information (FCI) field, which can be used for the feedback information that needs to be transmitted. Figure 4.5 shows this common packet format for feedback messages.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|   FMT   |       PT      |          length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of packet sender                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of media source                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:            Feedback Control Information (FCI)                 :
:                                                               :
```

Figure 4.5: Common packet format for feedback messages

The RFC also specifies a generic NACK Transport Layer Feedback message type, which is used to identify the sequence numbers of the packets that are considered lost. The Generic NACK message consists of two fields:

- **PID** (Packet ID) - this field contains the sequence number of the lost RTP packet
- **BLP** (Bitmap of following Lost Packets) - this 16 bit field allows for reporting the loss of any of the 16 subsequent packets following the sequence number provided in the PID field. The respective bit is set to 1 when the packet is considered lost, with the least significant bit denoting sequence number PID+1 (modulo 2^{16}) and the most significant bit denoting sequence number PID+16 (modulo 2^{16}).

For example, an Generic Nack message with PID = 0000001000000011 and BLP = 0000001000010111 reports the loss of the packet with sequence number 515, and of the 16 subsequent packets the packets with sequence numbers 516,517,518, 520 and 525 are also considered lost. This allows for efficient usage of the feedback messages when burst losses occur. A RTCP feedback packet can contain multiple generic NACK messages. This can be helpful when the detected packet loss length is longer then the 17 sequence numbers that can be reported with one Generic NACK. Whether it is useful to transmit multiple generic NACKs in one RTCP feedback message is arguable due to time constraints for retransmissions. When the application waits to transmit a RTCP packet to allow a wider range of packets that require retransmissions to be inserted in the same message might lead to a more efficient usage of RTCP packets, but the introduced delay also leads to a reduction of the time available for the succesful recovery of a packet.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            PID                |             BLP               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4.6: RTCP Generic NACK message

Summarizing, with one Generic NACK message a receiver is able to indicate the loss of up to 17 subsequent packets. Multiple Generic NACK messages can be added in one RTCP FB message, allowing the IPTV client to indicate the loss of even more packets.

Retransmission response

In section 2.5.3 a RTP retransmission payload format [37] has already been briefly. Because the IPTV packet retransmission mechanism does not require any application layer adaption, this payload format is suitable for required retransmission of the IPTV stream packets. The format of a retransmission packet is shown in figure 4.7.

The RTP header field contains a standard RTP header, with the following adaptations:

- The SSRC field contains the same synchronisation source used for sending the multicast IPTV stream. This allows the receiver to relate a retransmission packet to an IPTV stream stream.
- The sequence number field follows standard RTP rules; it must be one higher than the sequence number of the preceding packet sent in the retransmission stream.
- The retransmission timestamp must be identical to the timestamp of the original packet.
- The payload type is dynamic and can thus be determined by the application.

The payload of a RTP Retransmission packet consists of the retransmission header, which consists of a 16 bit field containing the original sequence number (OSN), and the original RTP packet payload (including any optional payload specific RTP headers). This is also shown in figure 4.7.

4.2.2 Retransmission protocol configuration

The previsously mentioned RTP protocol extensions provide the functionality for requesting retransmissions and the functionality for retransmitting packets. Other functionality is needed to provide a fast retransmission mechanism for a multicast IPTV service, which is not covered by these extensions:

1. Packet loss detection

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         RTP Header                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            OSN                |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                  Original RTP Packet Payload                  |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4.7: RTP Retransmission Payload Format

2. Retransmission feasibility
   (a) Expected packet recovery time
   (b) Expected packet read-out time
3. Multiple retransmission attempts per packet

1. Packet loss detection

Packet loss can be detected in several ways:

- By means of gaps in sequence numbers
- By means of the expected arrival time of a packet

The first approach requires regular transmission of RTP packets, because packet loss can only be detected upon the arrival of a packet subseding the packets(s) that are lost. In case of burst losses the burst loss will only be detected after the burst has ended, not during the burst, which decreases the changes to recover the lost packets in time.

The second approach detects loss by checking the expected arrival time of a packet. In a setup where the IPTV packets are send in a periodic manner, an IPTV client can estimate the arrival of a specific packet based on the arrival time of previous packets. This allows the client to determine that a packet can be considered lost if it has not been received after its expected arrival time. For this mechanism to operate correctly, the Streaming Sever needs to send the IPTV stream packets periodically. Clock synchronization between client and server is also preferred, but in the article by Wu and Liew (1999) a scheme is proposed which does not require clock synchronization [65].

As the second method is more complex and requires more processing power compared to the first solution, for the prototype the first approach is chosen.

2. Retransmission feasibility

To determine if it is feasible to recover a specific packet before the packet is needed, the following parameters play a role:

1. The packet recovery time: the time between sending a retransmission request and receiving the packet;
2. The moment in time the packet is expected by the application;
3. The maximum amount of retransmission requests per packet.

(a) Expected packet recovery time The packet recovery time depends on the IPTV client processing, the Round Trip time between the IPTV client and the retransmission cache, and the processing in the Retransmission Cache.

RTP session participant can determine the round trip time delay between sender and a RTP receiver by comparing a sender report with the receiver report (see [1], Section 6.4.1.). For RTP sessions in which certain members do not send, these RTP participants are not able to determine the round trip time delay, as sender reports, which are needed for round trip time computations, are not transmitted. This is the case in the IPTV multicast distribution network

As an alternative the expected packet recovery time is estimated based on previous packet recovery times. The IPTV client will calculate the time difference between sending a retransmission request and the reception of the respective retransmitted packet, i.e. the round trip time between a retransmission request and retransmission response. This measured time then gives an indication of the expected recovery time of the next retransmission request. To improve the accuracy of the expected recovery time a smoothing average is computed over previous successfully recoveries. This mechanism is similar to the retransmission timeout mechanism used by TCP [10].

The smoothed round trip time estimation *srtt* is given by:

$$srtt_{n+1} = \alpha * RTT + (1 - \alpha) * sRTT_n \tag{4.1}$$

where $0 \leq \alpha \leq 1$.

The round trip time variance estimation is given by:

$$rttvar_{n+1} = \beta * |RTT - sRTT_n| + (1 - \beta) * rttvar_n \tag{4.2}$$

where $0 \leq \beta \leq 1$.

This smoothed round trip time estimation and the variance of the round trip time are then used to compute the retransmission timeout (RTO), which is the expected recovery time for a packet:

$$RTO_{n+1} = \lambda * RTT + \gamma * rttvar_{n+1} \tag{4.3}$$

where $0 \leq \gamma \leq 1$ and λ is a constant.

The measured RTT values for a packet retransmission can only be used when only one retransmission request has been sent for a specific packet. When multiple requests have been transmitted, a ITPV client does not know which request lead to the the reception of the packet. This can lead to incorrect RTT value computations, as shown in figure 4.8. This algorithm is commonly referred to as Karn's Algorithm [66].

A drawback of this round trip time estimation technique is that in the case that no retransmissions have occurred or the the interval between retransmissions is quite large, the estimation of the round trip time might not be accurate. But because the Retransmission Cache is placed in an Access Node, fair assumptions of the maximum round trip time can be made.

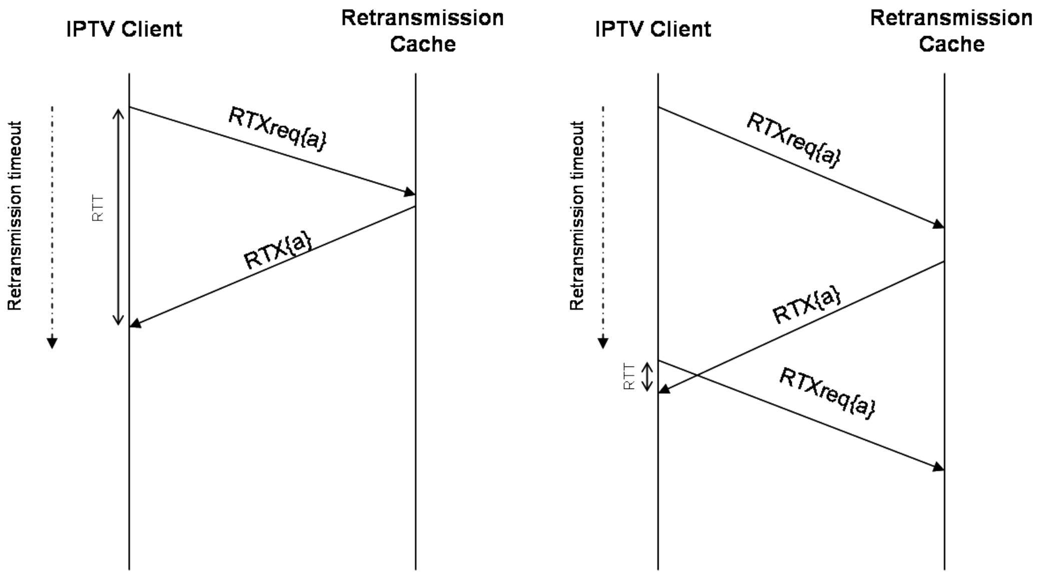

Figure 4.8: Retransmission RTT estimation (correct and incorrect)

(b) Expected packet read-out time A packet will be in the IPTV client packet buffer until the application layer needs to retrieve the packet payload for audio/video decoding. The time between arrival in the packet buffer and removal can be used as an threshold to determine whether a retransmission is possible, i.e. if the client should try to request retransmission of the packet, or not.

The expected amount of time before a packet will be read ($PktExp(i)$) can be estimated by dividing the distance of the packet to the head of the buffer with the rate at which the application reads from the buffer:

$$PktExp(i) = \frac{headOfBuffer - i}{applicationpacketreadrate} \tag{4.4}$$

When this expected read time is smaller than the expected recovery time, the IPTV client will ask for retransmission, otherwise the packet retransmission will likely be useless, so no attempt for recovery will be made.

3. Multiple retransmission attempts per packet

It can happen that a retransmission request is lost, or that a retransmitted packet is lost as well. This can be solved by resending a retransmission request when the IPTV client determines that a previous retransmission attempt has failed. The IPTV client can use a retransmission timeout to determine if a retransmission attempt has failed: when a retransmission request is sent, the retransmission timeout, as presented above, is used to create an expire timer. When this timer has expired, the previous retransmission attempt is considered failed. At this time, when additional retransmission requests are allowed, the feasibility of a new retransmission attempt is checked, and if succesful recovery is still possible, a new retransmission request is sent for the packet.

4.2.3 Transmission type

The IPTV streams are transmitted via multicast to the IPTV set top boxes of the users. For retransmissions two transmission types are possible: multicast or unicast delivery.

packet retransmissions can be sent either For retransmissions this however does not have to be the case. The retransmission packets can be sent either by multicast to all clients listening to the retransmission session for the specific channel, or by unicast to the specific client that requested the retransmission.

Multicast retransmission

Multicast delivery allows the retransmitted packet to be received by all the IPTV clients that might be interested in receiving the packet. This can be useful when there is a correlation between the packet losses for different IPTV clients. In this case the Retransmission Cache only sends one packet which will be delivered to all IPTV clients that are part of the multicast group. This means that only one retransmission request for a specific packet is required by the Retransmission Cache. When a IPTV client sends a retransmission request via unicast, other IPTV clients will not be aware of this requests. Several clients might thus be requesting the same packet, which can lead to feedback explosion or retransmission explosion.

To prevent the unnecessary transmission of packets (retransmission explosion) the Streaming Server may suppress retransmissions by maintaining a list of recently transmitted packets and does not allow a packet on this list to be retransmitted within a certain time interval. This however conflicts with allowing repeated retransmission attempts made by one client.

Alternatively the RTCP retransmission requests are transmitted to all IPTV clients by means of multicast RTCP feedback. Suppression of identical retransmission requests can then be provided by using a random back-off timer before sending a retransmission request; when this timer expires and the IPTV client has not received the same retransmission requests from another IPTV client, it will send the request, otherwise the request will be suppressed. This mechanism is also part of the extension to RTP allowing for early RTCP based feedback, discussed in section 2.5.2.

Because the retransmission request messages need to be transmitted to all clients, multicast feedback can lead to a high increase in bandwidth usage for each access link, even when feedback suppression is enabled. But more important, the multicast retransmissions self can lead to problems: in a worse case scenario, all packets need to be retransmitted because IPTV clients all request different packets, doubling the bandwidth consumption for IPTV stream delivery. This can happen when in a large multicast group IPTV clients all ask for the retransmission of different packets. This can result in one retransmission packet per IPTV stream packet. The amount of bandwidth required for the multicast retransmissions depends on:

- the number of IPTV clients that are part of the multicast group;
- the packet loss characteristics.

A different concern, which was already addressed in chapter 2 is that IPTV clients might not send multicast packets due to privacy concerns or network concerns.

Unicast retransmission

In a unicast setup the retransmission requests are sent directly to the Retransmission Cache (or snooped by the Retransmission Cache) and the Retransmission Cache send the packets directly to the IPTV client. This retransmission mechanism operates independently of other IPTV clients that might request retransmissions. The suppression mechanisms that are needed in a multicast setup are not needed in a setup with unicast retransmissions, which thus reduces the complexity of the solution significantly. But this also means that the Retransmission Cache may have to send the same packet multiple times when multiple clients request the same packet.

In a worst case scenario a packet has to be transmitted to every IPTV client. Therefore the processing performance of the Retransmission Cache can become an issue, depending on how many IPTV clients need to be supported.

Conclusion

When comparing the two different approaches the following conclusions can be drawn:

- For multicast retransmissions the capacity of the access links becomes a bottleneck when the number of IPTV Clients with packet loss problems increases. Because feedback suppression is needed the solution needs to support these mechanisms, making the IPTV client and Retransmission Cache more complex. Furthermore are retransmissions delayed because of the required suppression mechanism, which reduces the time window for a successful recovery.

- For unicast retransmissions the processing performance of the retransmission cache can become a bottleneck when the number of clients that asks for retransmission increases, but the solution requires less complexity.

Based on this comparison the prototype will be using a unicast retransmission scheme.

4.2.4 Retransmission protocol parameters exchange

For the retransmission mechanism to be functional it is necessary that the IPTV client and Retransmission Cache agree on certain configuration parameters:

- the destination address (IP-address and port number) of the retransmission stream;
- the network address of the retransmission cache;
- the available bandwidth for packet retransmissions;
- the time packets will be available for retransmission.

The exchange of these parameters will typically occur before or during the setup of a streaming session using one of the protocols discussed in section 2.4. The design and implementation of the exchange and negotiation of the parameters related to setting up the retransmission is out of the scope of the prototype. However a short explanation of how the parameters would be exchanged is explained below.

The session description should, additionally to the session related to reception of the IPTV stream, contain a identifier to state that retransmissions are supported, the address which is used for packet retransmissions, the retransmission payload format, and the time packets are available for retransmission. RFC 4585 [36] introduces new MIME-types that can be used to exchange parameters related to RTP packet retransmission.

MIME TYPE	Description
application/rtx	Retransmission Session
rtx-time	Indicates the time in milliseconds (measured from the time a packet was first sent) that the sender keeps an RTP packet in its buffers available for retransmission.

Table 4.1: MIME-types for retransmission parameter exchange

4.3 Prototype implementation

In this section the implementation of the prototype is discussed. The following three sections discuss the details regarding the tree implemented system components and section 4.3.4 provides the configuration details of the prototype and its restrictions.

4.3.1 IPTV Streaming Server

The only purpose of the Streaming Server is to provide IPTV channels that are multicast using RTP over UDP. The server continuously sends an IPTV packet stream to a multicast group address. There are numerous software packages available that offer RTP based video streaming. One of the most versatile programs currently availabiel is the Videolan VLC Media Player [67]. VLC offers streaming of a large number of audio and video formats using various transport protocols, including UTP and RTP. VLC also supports multicast distribution and the MPEG-TS format. Therefore VLC is used as the Streaming Server in the prototype setup.

4.3.2 Retransmission Cache

The Retransmission Cache is implemented in Java and uses the Java RTP API created by Waqar Ali and Akhil Nigam [68]. To support packet retransmissions, this API is extended to support the RTCP Generic NACK retransmission requests and the RTP retransmission payload format. The basic application logic is provided to process retransmission requests and a packet buffer is implemented for the storage of RTP packets. The packets are stored in a packet buffer with a predefined buffer size and are placed and removed from the buffer using a first-in-first-out (FIFO) approach: when the packet buffer is full, the oldest packet will be removed and therefore not available anymore for retransmission.

Figure 4.9 gives a simplified overview of the Retransmission Cache implementation, by means of an UML activity diagram. What can easily be seen is that the Retransmission Cache functionality is quite restricted, resulting in a software program with little complexity. This is an advantage for a component that might be used in an access node, which has limited processing power.

4.3.3 IPTV Client

The IPTV client is the most complex component of the prototype: it has to support IPTV streaming, decoding of audio and video packets and the retransmission functionality. The IPTV client is implemented in C using several libraries, which will be explained below.

Transport layer functionality

For providing RTP support, the C RTP API developed by Schulzrinne and Lennox, RTPLIB is used [69]. This library is also extended to support the functionality required for retransmissions: the Generic Nack message, the common feedback RTCP packet format and the RTP Retransmission payload format. With

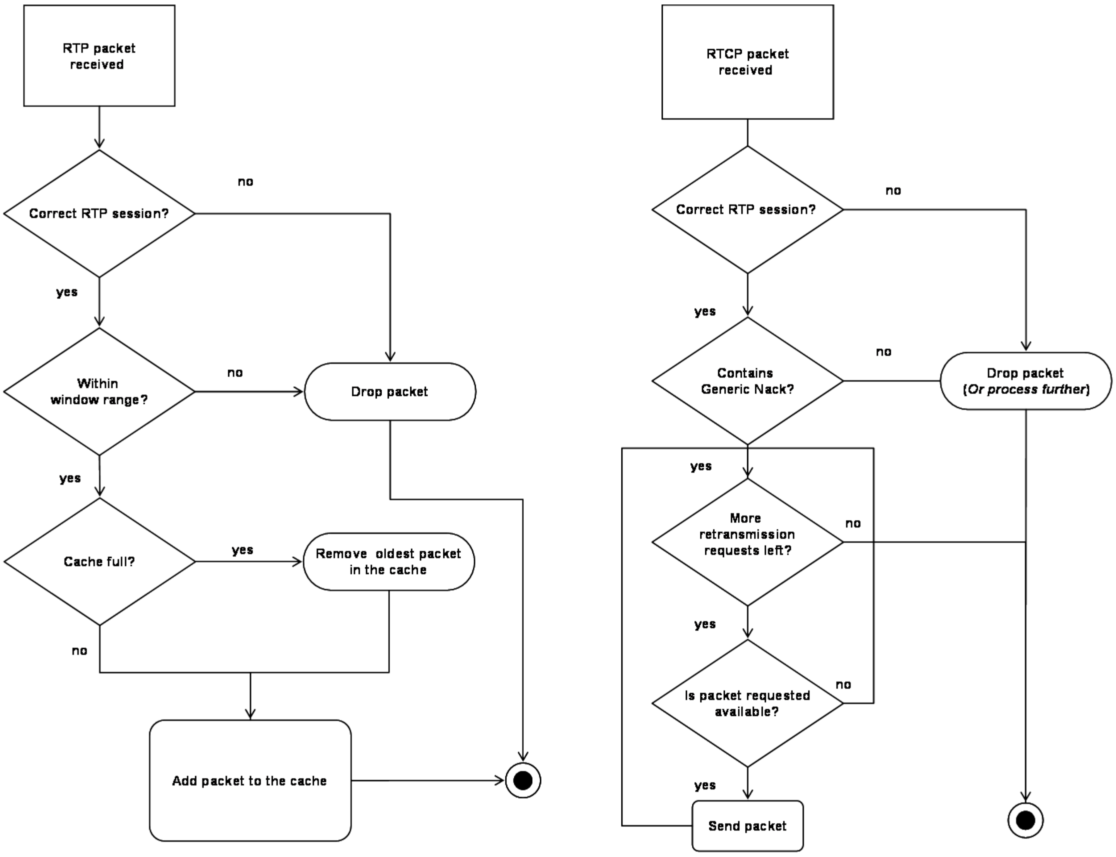

Figure 4.9: Retransmission Cache activity diagram

this extended library an application is created which recieves an IPTV RTP stream, stores packets in a packet buffer, determines packet loss, asks processes retransmitted packets.

A simplified overview of the IPTV client implementation is given in the activity diagram described in 4.10. This diagram contains the transport layer functionality, without the retransmission timeout mechanism. The figure shows that upon reception of a RTP packet several steps are taken to put the packet in the buffer and, when packets are missing, ask for retransmission. It also shows how upon the reception of a retransmission packet the packet is placed in the buffer.

Presentation and application layer functionality

The media player parts of the IPTV client prototype are implemented with the FFMPEG [70] and SDL [71] libraries.

FFMPEG is a large collection of multimedia libraries which can be used to encode and decode multimedia in different audio and video formats. The library consists of two parts:

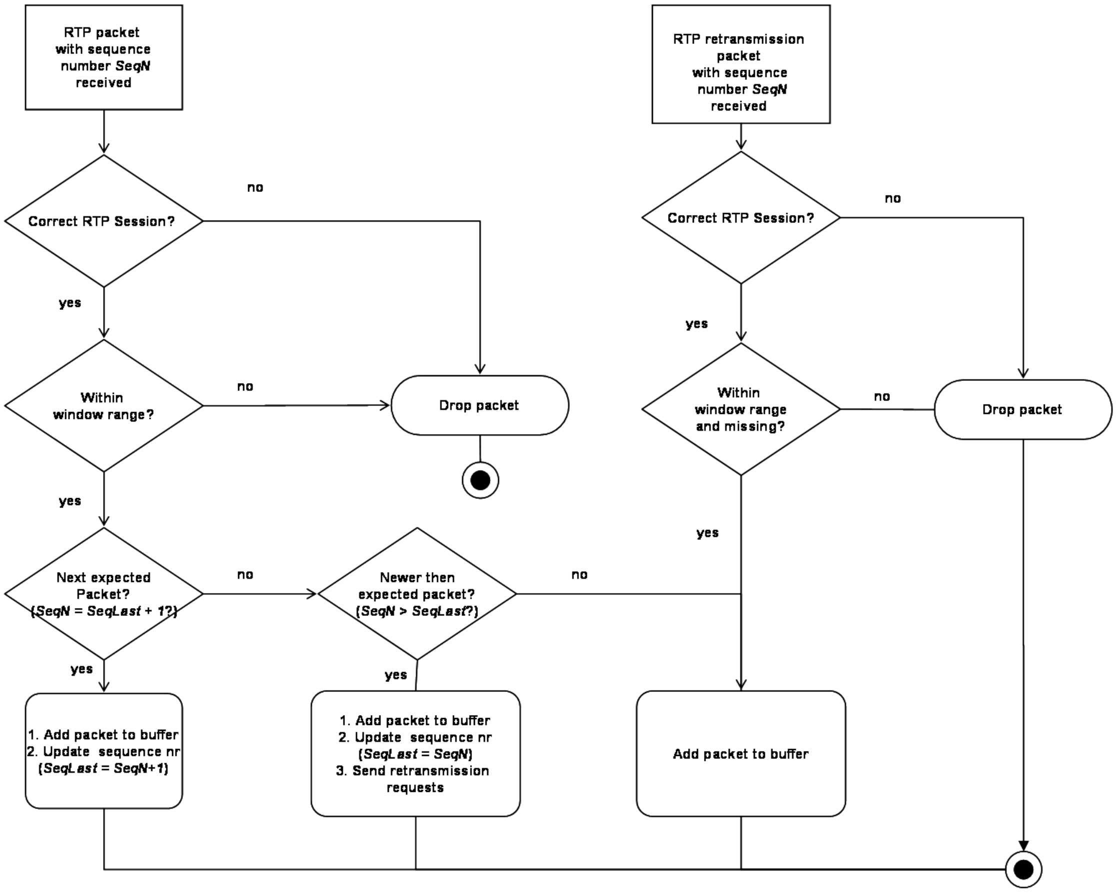

Figure 4.10: IPTV Client activity diagram

- libavformat - a multimedia library which can multiplex and demultiplex multimedia formats;
- libavcodec - a large collection of open source audio and video codecs.

The libavformat library is used to demux the audio and video streams in the MPEG-TS stream. These respective streams are then decoded using the MPEG-2 video and MP2A audio decoders provided by the libavcodec library.

The open Simple DirectMedia Layer (SDL) library is used to create a simple media player GUI which outputs the video and audio data decoded with libavcodec.

The network layer and application layer components are used to create a prototype IPTV client, which can receive decode and display IPTV streams and use the introduced retransmission functionality to recover missing packets.

4.3.4 Prototype configuration

In this section the most important configuration aspects of the implemented prototype are given. First the configuration of the Retransmission Cache and IPTV client are given, then the details of the retransmission mechanism function are discussed and the limitations of the prototype are given.

Retransmission Cache The Retransmission Cache configuration does not deviate from the design: the IPTV stream packets are placed in the buffer and when the buffer is full the oldest packet will be dropped. When a RTCP FB message is received, for each retransmission request a retransmission packet is sent, if the packet is still available in the buffer. The packet is sent to the IPTV client requesting the retransmission. Other RTCP messages like receiver reports are not processed any further. The Retransmission Cache supports the reception of RTCP transmitted to the Retransmission Cache directly, but also RTCP packets that are transmitted to another RTP participant (the Streaming Server). In this case the packets are "snooped".

IPTV client The IPTV client is configured as follows: when packet loss is detected, the difference between the last received packet and the packet currently received determines how many packets are considered lost. A RTCP feedback packet contains exactly one Generic Nack message, which can contain requests for up to 17 packets, as explained in section 4.2.1. When more than 17 subsequent packets are missing, a new RTCP feedback packet will be created for the packets missing after the 17th packet (i.e. two RTCP feedback messages are sent to the Retransmission Cache). The RTCP packets are transmitted directly to the Retransmission Cache, so the snooping functionality of the prototype is not used.

The RTCP feedback packets are transmitted immediately, such that the packet recovery time is kept to a minimum. Because the RTCP feedback packets are sent immediately the Generic Nack message contained in the feedback packet may not be used efficiently when the packet loss does not occur in bursts. This will be explained with the following examples:

When a IPTV client receives a packet with sequence number *X* and subsequently receives the packet with sequence number *X+6*, the application detects the loss of 5 packets (*X+1* up to *X+5*). The resulting Generic Nack message will thus contain retransmission requests for 5 packets.

When a IPTV client receives packet *X* and subsequently receives packet *X+2*, the application detects the loss of packet *X+1* and the resulting Generic Nack will only contain one retransmission request. When the client now receives packet *X+4*, a new Generic Nack message is created for packet *X+3*.

When the transmission of the first Generic Nack message would have been delayed, then the request for packet *X+3* could have been added to this Generic Nack message, which means that a second RTCP feedback packet does not have to be sent.

A possible solution thus would be to create a balance between delaying the transmission of a RTCP feedback packet and waiting for the loss of a packet that would result in a retransmission request that could be inserted in the delayed feedback message. Due to a lack of time this feature was not implemented.

Repeated retransmission attempts When a retransmission requests is expired and the application is configured to allow multiple retransmission attempts, a new Generic Nack message will be created. As multiple retransmission request for different packets may have been expired, when possible, these are also combined in one Generic Nack RTCP message. For retransmission retry attempts the Generic Nack message may thus also contain up to 17 retransmission requests.

Transmission constraints The transmission of the RTCP feedback messages occurs regardlessly of the RTCP transmission interval, i.e. there is no constraint on the number of RTCP feedback messages that the IPTV client may send. An advantage of this approach is that for all missing packets a retransmission attempt is allowed; a drawback of this approach is that it can lead to a lot of feedback messages being sent when loss rates are high. For access links with limited bandwidth resources this may cause a problem, because in the uplink direction a lot of feedback messages may be transmitted, while in the downlink direction a lot of retransmission packet retransmissions may occur.

Packet retransmission feasibility The check for feasibility of retransmissions is not enabled in the prototype, because early runtime results showed that this mechanism was not accurate enough, leading to the suppression of retransmission attempt, while there was sufficient time for successful packet recovery. Unfortunately there was no more time to improve the algorithm, so the functionality is disabled in the prototype.

Retransmission timeout detection During early tests with the prototype implementation it became clear that a lot of duplicate retransmissions occurred. Analysis of the test results showed that retransmission requests were considered lost too early, as retransmission packets were received after the retransmission timeout expired. This resulted in unnecesarry repeated retransmission requests and packet retransmissions. Further analysis showed that the computed expected round trip time showed high amounts of variability. This was caused by restrictions of the prototype implementation: the computed retransmission request transmission times and retransmission packet arrival times were influenced by application processing. This means that the reported times also included the processing by the IPTV client, which added variability to the estimated average round trip times. This problem was solved by setting λ to 2, in the retransmission timeout formula:

$$RTO_{n+1} = 2 * RTT + 1.5 * rttvar_{n+1} \tag{4.5}$$

Retransmission configuration parameter exchange As stated earlier, the exchange of configuration parameters between the Retransmission Cache and IPTV client is out the scope of this thesis. Therefore configuration of the Retransmission Cache and IPTV is done statically.

Chapter 5

Prototype evaluation

In the previous chapter the design and prototype implementation of the fast retransmission mechanism for multicast IPTV were presented. In this chapter the prototype will be evaluated to determine the minimum requirements for providing retransmissions and the applicability of the fast retransmission mechanism under different network environments. The chapter consists of the following:

In section 5.1 the experiments are described that are used for evaluation of the fast retransmission mechanism. In section 5.2 the experiment measurement methodology is explained. Section 5.3 provides the experiment setup and discusses the tools that are used during the experiments. Sections 5.4 to 5.6 discuss the experiment results. Finally some insights about the applicability and scalability of the solution are provided in section 5.7.

5.1 Performance experiments

In Chapter 1 the following research question was presented:

Under which (network) scenarios can RTP-based packet retransmission be successfully applied as a error resiliency mechanism (for multicast IPTV TV channels)?

To answer this question the performance of the retransmission mechanism is evaluated in a network setup with one IPTV client. Because there are a lot of aspects that can influence the performance of the fast retransmission mechanism, three experiments are conducted to achieve the following goals:

1. Determine the minimum required buffer size for the IPTV client and the Retransmission Cache respectively for the retransmissions mechanism to be beneficial. This means, the retransmission mechanism leads to the reduction of application payload losses and therefore an improvement of the IPTV service.

2. Determine the applicability of packet retransmission under different loss scenarios in an uncongested access network.
3. Determine the applicability of packet retransmission under different loss scenarios in a congested access network.

5.1.1 Justification

The first goal focuses on investigating the minimal requirements for the packet retransmission mechanism to become applicable. This is determined by looking at the dimensioning of the buffer for both the Retransmission Cache and the IPTV client. The results of this experiment provides valuable information because of the memory limitations of hardware ITPV clients and access node equipment (e.g. a MSAN or a DSLAM).

For successfull application of the retransmission mechanism, the Retransmission Cache must have a minimum buffer size and the IPTV client must have a minimum prebuffer size (the amount of data buffered before playback starts). The buffer sizes are primarily determined by time needed to perform a succesful retransmission. This time consists of:

1. The time between the detection of packet loss and the scheduling of the request.
2. The propagation delay from client to cache.
3. The time between the reception of the retransmission requests and the scheduling of the retransmission packet.
4. The propagation delay from cache to client.
5. The time it takes to process a retransmission packet and place in in the queue.

In the experiment an attempt is made to discover the size of the Retransmission Cache and the size of the client buffer to allow for a retransmission request to be fulfilled and to allow a retransmission to occur, before the packet is needed by the application for decoding and displaying.

The second goal focuses on investigating the effect of different network conditions on the performance of the packet retransmission mechanism. It is expected that different loss scenarios lead to different packet loss recovery results. In this experiment the focus lies on determining the applicability of retransmissions that occur due to network errors, not due to congestion. The utilization of the access link will therefore will kept below the maximum capacity, such that cognestion does not occur. This experiment also give insight about the retransmission bandwidth overhead under different network environments.

The third goal focuses on investigating the effects of the retransmission mechanism when packet loss is caused due to network congestion. In a IPTV distribution network with a congested access link or a access link that is likely to get congested, packet retransmissions may lead to a decrease of the IPTV service performance, as the generated retransmission traffic can contribute to congestion. In this experiment a large FTP file transfer will be used to create additional traffic, leading to congestion in the access network.

5.1.2 Performance metrics

In the experiments several metrics will be used for performance evaluation. The metrics that are used are presented in table 5.1.

Definition	Description
Uncorrected packet loss rate (PLR)	Percentage of IP packets lost in the network before applying error recovery
Corrected packet loss rate (PLR)	Packet loss rate after correction by packet retransmission
Duplicate packet rate	Percentage of retransmission packet received multiple times
Retransmission bandwidth ratio	The ratio of bandwidth required for packet retransmission compared to the bandwidth needed for the IPTV stream
Cache hit ratio	The percentage of retransmission requests that can be fulfilled by the retransmission cache.
Retransmission success ratio	The ratio of missing packets that are recovered with retransmissions and are usable by the IPTV client.
Delayed frames ratio	The ratio of video frames that are displayed after their scheduled presentation time.

Table 5.1: Prototype evaluation metrics

The uncorrected and corrected packet loss rate show the recovery performance of the retransmission mechanism, as the difference between the two rates show the performance improvements that are achieved by enabling packet retransmissions. These metrics are part of the proposed QoS/QoE metrics standards provided by the IETF focus group on IPTV, as discussed in section 2.12.

The duplicate packet rate metric gives insight in the configuration of the retransmission request timeout mechanism; when the duplicate packet ratio is high, the retransmission timeout interval may have been set too small.

The retransmission bandwidth ratio gives insight in how the retransmission packets bandwidth requirements relate to the IPTV stream and thus can help to determine if applying packet retransmission is suitable in a access network with a limited capacity.

The retransmission cache hit ratio and retransmission success ratio are used to determine the buffer sizes of the retransmission cache and the IPTV client respectively. When the respective buffer sizes are too small both ratios will be zero. When the buffer sizes increase both values will increase to (at most) 100%, which thus allows to determine the minimum required buffer sizes.

The delayed frames ratio is used to determine if the IPTV client is able to display the video stream fluently. This is related to the amount of video data that needs to be buffered before fluent playback is possible. When the amount of buffered video data is too low, the video decoder cannot decode and display the video frames at the right time, which results in jerky video playback. The delayed frames ratio gives an indication if the selected prebuffer size is large enough for fluent playback, or that the buffer is to small such that fluent playback is not possible.

To be able to determine these metrics, the Retransmission Cache and IPTV client measure certain parameters during an experiment. An overview of the parameters that are measured is provided in Appendix A. After the experiment these parameters are then used to compute the metrics.

5.1.3 Experiment configuration parameters

The experiment configuration parameters are provided in table 5.2. These parameters provide the settings for the experiments, including the variables that are being used for evaluation. Besides containing the the network configuration settings the settings include application configuration paramters and settings related to the duration of the experiments.

Parameter	Definition
Downlink delay	The amount of delay (in ms) introduced in the network in the downlink direction
Downlink loss	The amount of packet loss introduced in the network in the downlink direction (in %)
Uplink delay	The amount of delay (in ms) introduced in the network in the uplink direction
Uplink loss	The amount of packet loss introduced in the network in the uplink direction (in %)
Access link capacity	The capacity of the access link
Client buffer size	The fill level of the buffer (in bytes), before playback is started
IPTV stream bit rate	The bitrate of the IPTV stream
IPTV stream IP packet size	The size of an IPTV IP packet (in bytes)
IPTV retransmission packet size	The size of a retransmission IP packet (in bytes)
Retransmission Cache buffer size	The size of the Retransmission Cache buffer (in bytes)
Experiment duration	The duration of the experiment
FTP file transfer duration	The duration of the FTP file transfer introduced in the network

Table 5.2: Experiment configuration parameters

5.2 Experiment measurement methodology

For the experiments the IPTV client will run for a fixed amount of time. In this time-interval several parameters are gathered:

- The number of RTP packets received from the Streaming Server (the IPTV stream)
- The number of RTP packets that are considered missing (before any retransmission attempt)
- The number of RTP packets actually missing after recovery by packet retransmissions
- The number of RTP packet received from the Retransmission Cache (the retransmitted packets)
  - The retransmitted packets received that can still be used by the application
  - The retransmitted packets received that are discarded because they are duplicate or received too late

During execution the following values are continuously monitored

- The number of packets is the packet buffer. This value is used to continuously update the average fill size of the packet buffer.
- The round trip time delay for a retransmission request and a retransmission response, used to determine the retransmission request timeout interval.

After execution an experiment run, the metrics are computed.

5.2.1 Reliability and confidence estimation

To be able to give confident conclusions based on the experiment results, each experiment run is repeated several times, using the same experiment settings. The measured sample values from the experiment runs are used to compute the mean value of a metric with confidence limites for this mean. The confidence limits of the mean are computed using a Student test or t-test [72]. The confidence limit is defined as:

$$\overline{\Upsilon} \pm t_{\alpha/2,N-1} S / \sqrt{N} \tag{5.1}$$

where $\overline{\Upsilon}$ is the sample mean, S is the sample standard deviation, N is the sample size, α is the desired significance level, and $t(\alpha/2, N-1)$ is the upper critical value of the (Student) t distribution with $N-1$ degrees of freedom. In the following experiments a confidence interval of 95% will be used.

5.2.2 Experiment execution plan

The experiments are performed in batches, which means that the experiment is repeated with the same settings to be able to provide reliable results. After an batch has been performed, the experiment parameters are changed an the experiment is repeated. This is done until all parameter settings have been tested. The following execution plan is used as a guideline:

1. Initialization of the experiment
   (a) Initialization of the test network
   (b) Initialization of the Streaming Server
   (c) Initialization the IPTV Client
2. Exetution of the experiment
   (a) Run an experiment batch
      i. Run the experiment for the predetermined time
      ii. Store the experiment results
      iii. Repeat the experiment until all batch runs are completed
   (b) Change the parameters of the experiments
   (c) Repeat the batch with the new settings
3. Processing and evaluation of the results

5.3 Experimental setup

For the experiments the prototype implementation is installed in a test setup. The test setup is used to emulate a multicast IPTV distribution network, consisting of a IPTV Streaming Server, a Retransmission Cache and a IPTV client.

5.3.1 Hardware inventory

The prototype components are installed on standard PC systems running Ubuntu Linux which are placed in a test network environment that resembles a IPTV distribution network.

The following hardware components are used:

- One system for the Streaming Server
- One system for the IPTV client and the traffic generator client for the reception of the TCP stream
- One system for (access node) routing, the Retransmission Cache and traffic generator
- One system for network emulation

5.3.2 Network topology

In figure 5.1 the topology of the test network is shown. The setup consists of two separate networks, depicted as ‘Subnet A‘ and ‘Subnet B‘. Subnet A reflects the IPTV service provider network. It contains the Streaming Server, which streams an IPTV stream using multicast RTP, and the Retransmission Cache, the device that temporarily buffers packets sent by the Streaming Server toward the IPTV clients. This sytem also contains an IGMP proxy, which is required for the routing of the multicast data generated by the Streaming Server. Subnet B reflects the access network and home network. It contains an IPTV client and the network emulator. The IPTV client receives the IPTV stream and displays the content on a screen. The Network Emulator is used to emulate different network conditions in Subnet B.

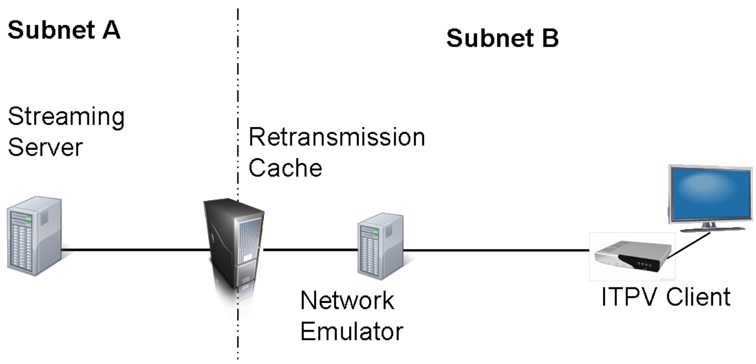

Figure 5.1: Testbed setup

Network load and throughput constraints

For the first two experiments the throughput of the network is not a constraint. The capacity of all network links is 100MBit/s. For the simulation focusing on congestion the network capacity in the network emulator will be limited. This is done to avoid creating additional throughput bottlenecks in the network that could influence the experiment. The network is only used by the applications that are required for the prototype evaluation. The transmitted data consists of:

- the RTP stream;
- RTCP packets (receiver reports and generic feedback messages containing retransmission requests;
- The RTP retransmission packets;
- TCP traffic generated by the traffic generator.

5.3.3 Network emulation

The experiments require the emulation of several network characteristics: the available bandwidth, the network propagation delay and packet loss. This functionality is provided by three different tools:

- TC, the traffic control package of the Linux kernel;
- Netem, a Network Emulator for the Linux kernel;
- TCN, Trace Control for NETEM.

TC is used to provide traffic control in the Linux kernel. It for instance provides different queuing mechanisms and allows for filtering packets based on specific parameters. In the experiments TC is used to filter the packets related to the IPTV streaming from all other (optional) traffic. This will ensure that the network emulation functionality provided by NETEM and TCN only affects the above mentioned data. Furthermore does TC allow to limit the throughput of a certain data flow, which is used in the third experiment for the creation of congestion.

To emulate network delay and loss Netem [73] is used. Netem is a software package that provides network emulation functionality for testing protocols by emulating the properties of wide area networks. It provides functions to emulate network delay, packet loss, packet duplication and packet re-ordering.

Netem does however not provide sophisticated functions to emulate burst packet losses. This functionality however is provided by TCN [74], which allows Netem to be used in combination with a trace file. The trace file contains delay and loss charactaristics on a per packet basis. This feature is used to emulate correlated (burst) packet loss.

Access link emulation For the experiments Netem is used to create network delay characteristics that resemble a DSL subscriber line: for the downlink direction a delay of 10 ms while be introduced; the uplink delay will be set to 2 ms.

Isolated loss emulation Isolated packet loss typically occurs due to bit errors occurring during transmission. To emulate isolated loss the packet loss function of the NETEM package is be used which randomly drops packets with a specified probability. The packet drop probability ranges from 2^{32} = 0.0000000232% up to 100%. Optionally a correlation may be added to allow the randomly dropped, which causes the random number generator to be less random. This feature is not used in the experiments.

Burst loss emulation A well know model for simulating burst packet loss is the Gilbert model, which is a Markov model. A Gilbert model consists of two states; one state with low loss rates (state 'G' or Good) and a state with high loss rates (state 'B' or bad). In table 5.2 the pseudo code for determining if a packet is lost is given, while in figure 5.3 the model is presented as a Markov state transition diagram. The Gilbert-model is used to create trace files for TCN.

```
% Gilbert model with states 1 and 2:
if (rand() < loss_probability[state])
{
loss = TRUE
}
else
{
loss = FALSE
}
if (rand() < transition_probability[state])
{
state = 3 - state
}
```

Figure 5.2: Gilbert-model pseudo-code

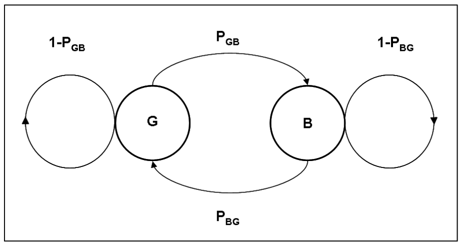

Figure 5.3: Two state Gilbert Model

Congestion emulation To emulate congestion two actions are taken: first the capacity of the access link is reduced by limiting the throughput with TC. The bandwidth is limited in such a way that the IPTV stream can be transmitted without interruption, but when additional traffic is in the network, congestion may occur. When congestion occurs packets that do not fit in the queue are dropped.

Traffic generation Additional traffic is generated by a traffic generator which will lead to congestion in the network emulator. To generate this traffic the Distributed Internet Traffic Generator (D-ITG) [75] software package is used. This package can be used to generate traffic for different protocols (e.g. UDP, TCP, or RTP) and using different inter-departure time models, like an exponential or poisson distribution.

5.3.4 IPTV Stream

The ITPV stream that is transmitted consists of a prerecorded movie with a audio and a video stream. The video is MPEG-2 encoded content, with a frame rate of 25 fps and a resolution of 704 by 576 pixels, which is a standard definition format. The audio stream is encoded in the MPEG-2 audio format. It consists of a stereo audio stream with a sample rate of 48khz and a bit rate of 192kbit/s.

In figure 5.4 the bandwidth usage of the IPTV RTP stream is shown. The RTP stream has an average bandwidth usage of 3.6 Mbit/s, with a minimum of 2.6 Mbit/s and a maximum of 4.5 Mbit/s. All IP packets have a size of exactly 1356 bytes, 1326 bytes for RTP, 8 for UDP and 20 for IP. One RTP packet contains 7 MPEG-TS packets of each 188 bytes.

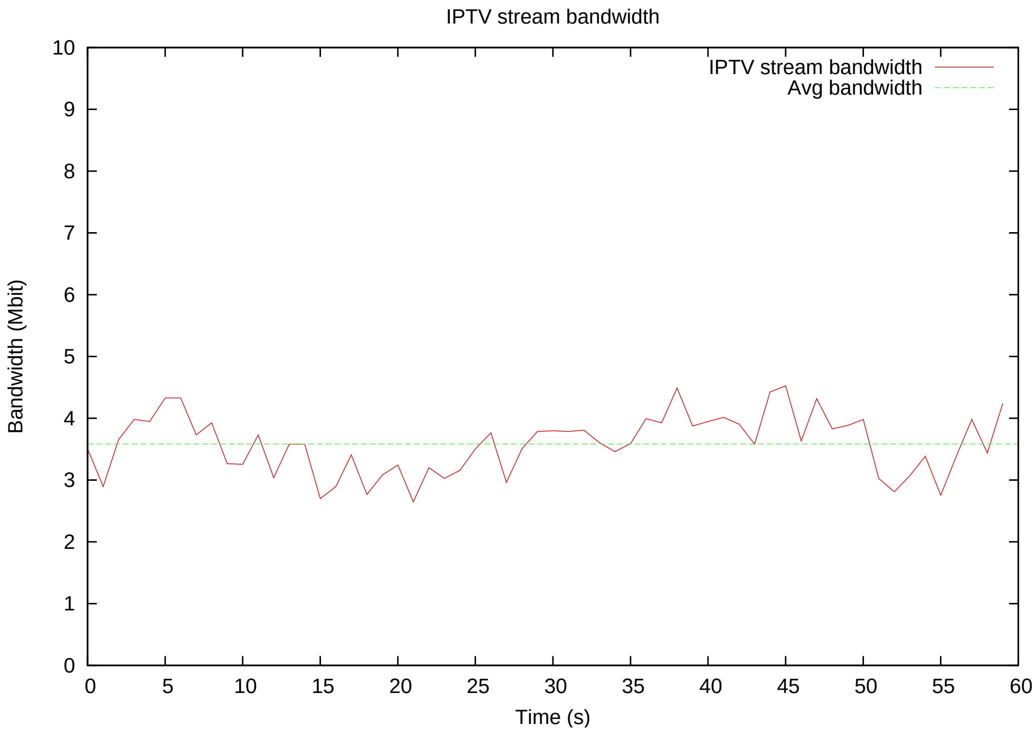

Figure 5.4: IPTV stream bandwidth usage over time

5.4 Buffer dimensioning

To determine the minimum buffer size requirements for the Retransmission Cache and the IPTV client to allow packet retransmissions, the retransmission mechanism is tested using different Client and Cache buffer sizes. The first part of the experiment focuses on the Retransmission Cache buffer, the second part focuses on the buffer of the IPTV client. In figure 5.5 the network setup displayed, including the data flows in the network, It also shows which data is affected by packet loss.

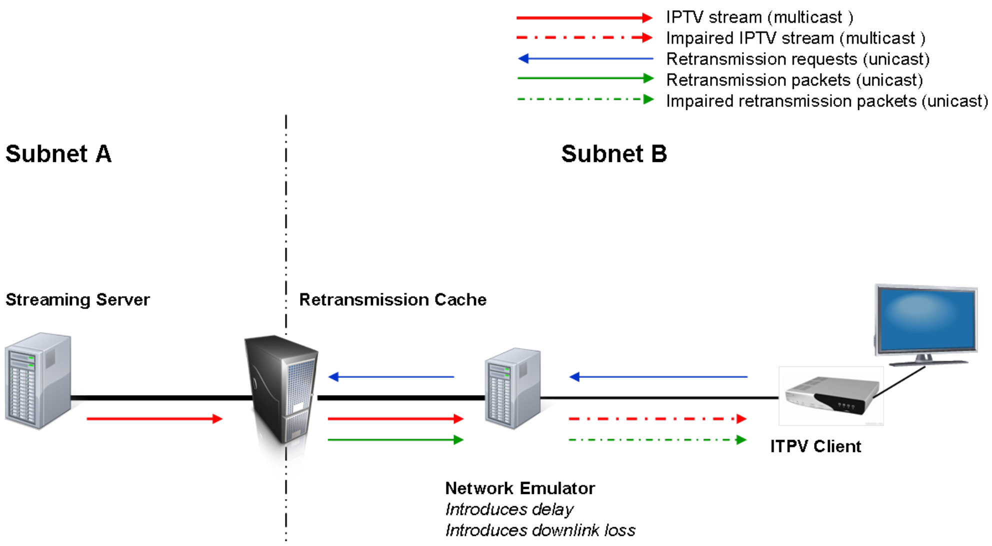

Figure 5.5: Network with an uncongested access link

5.4.1 Buffer dimensioning Retransmission Cache

The Retransmission Cache buffer size determines whether a retransmission request from an IPTV client can be fulfilled or not. When the buffer size is too small, the packets will already be removed from the buffer before a retransmission request is received by the cache. When multiple request for the same packets are allowed, the packet should still be available for subsequent requests. In the experiment the Retransmission buffer size is increased sequentially, while the IPTV client buffer size is kept constant. The buffer client size is predetermined and configured in such a way that a retransmissions can be applied successfully; so the only constraint influencing the applicability of packet retransmission is the Retransmission Cache buffer size.

During a experiment run the number of retransmission requests that can be fulfilled by the cache is counted, as well as the number of missing packets that actually get recovered in the IPTV client. The expectation is that when the buffer size is too small, the ratio of requests that can be fulfilled will be low and when the buffer size is increased more retransmission requests can be fulfilled until a limit is reached, where increasing the buffer size does not lead to a increase of the number of retransmission requests that can be fulfilled.

Each experiment run is executed multiple times and after these runs the Retransmission Cache buffer size is increased and the experiment is repeated. In this experiment the loss characteristics are fixed.

From the experiment result results two relations can be established:

- The client buffer size versus the packet discard ratio.
- The cache buffer size versus the packet not available ratio.

When these values are plotted in two graphs, an indication can be made of the minimum Retransmission Cache buffer size, to allow for successful application of the retransmission mechanism.

Experiment Settings

For the experiment the following settings are used:

Downlink delay:	10 ms
Downlink loss:	5% (uncorrelated loss)
Uplink delay:	2 ms
Uplink loss:	0%
Client buffer size:	1500 KB
IPTV stream:	3.6 Mbit/s
IPTV stream IP packet size:	1356 (fixed)
IPTV retransmission packet size:	1358 bytes
Retransmission Cache buffer size:	0-68 KB
Experiment duration:	120s

Results

Figure 5.6 shows the percentage of retransmission requests that can be fulfilled as a function of the Retransmission Cache buffer size. The presented values represent the mean values with a 95% confidence limit. The results clearly show that when the buffer size is 30 KB most retransmission requests can be fullfilled. The percentage of retransmission requests that can be fulfilled lies between 99 and 99.5%. When the buffer size is smaller than this value, the packets will not be available long enough to 'compensate' for the propagation delay between the Retransmission Cache and the IPTV client, the processing in the IPTV client (detection of loss, requesting a packet), and the propagation delay from the IPTV client to the Retransmission Cache. When the buffer size is 40 KB or bigger, all retransmission requests can be fulfilled.

The figure also shows the percentage of packets requested by the IPTV client that result in a successful recovery of the missing packet before the packet is needed. As what can be expected, the results clearly resemble the results perceived in the Retransmission Cache. When the buffer size is at least 40 KB the success ratio lies between 94.69% and 95.56%. The reason why this ratio does not increase to 100% is that some retransmitted packets are also lost.

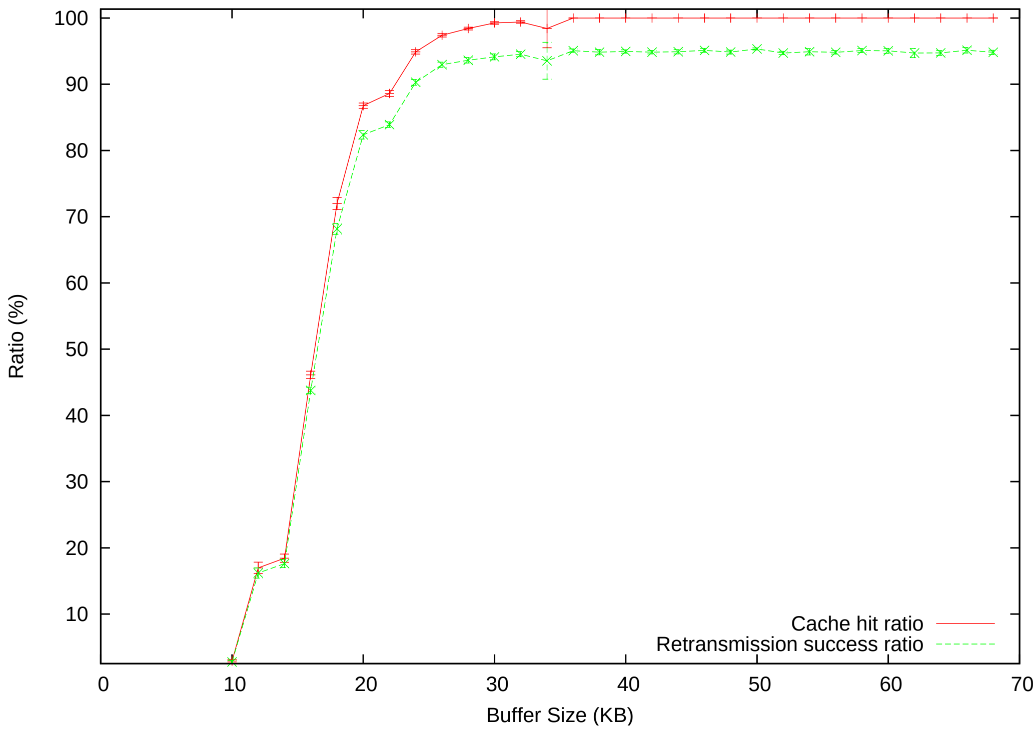

Figure 5.6: Buffer dimensioning Retransmission Cache with one retransmission attempt

Figure 5.7 provides the results for the same experiment, but now with the number of retransmission requests increased to at most three. The graph shows a similar relation between the the Cache hit ratio and the Retransmission success ratio, but has the following differences:

- The cache hit ratio with a packet buffer size of 30 KB is circa 13% lower compared to the previous result. This is caused by the repeated retransmission request for packets that cannot be fulfilled: the packet was not not available anymore at the time when the first retransmission request occurred; subsequent requests can then neither be fulfilled. The minimum required packet buffer size is now at least 60 KB, an increase of 50%.
- The retransmission success ratio improves when multiple retransmission attempts are used: in the experiment with only one retransmission attempt, the best available ratio was circa 95%; in this setup the ratio increases to around 99% when the packetbuffer size is at least 60 KB.

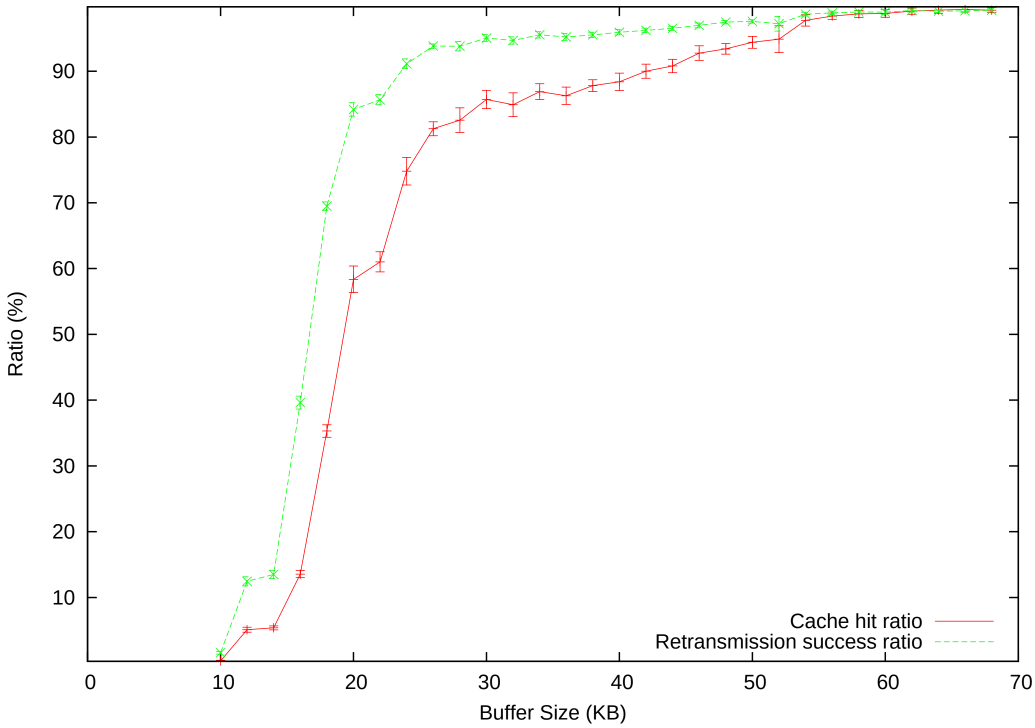

Figure 5.7: Buffer dimensioning Retransmission Cache with up three retransmission attempts

5.4.2 Buffer dimensioning IPTV client

The evaluation of the dimensioning of the buffer of the IPTV client is more complex than the evaluation of the buffer of the Retransmission Cache. Not only does the packet buffer size determine whether retransmissions are feasible, it also determine whether the IPTV application is capable of decoding the video frames and displaying them at the right time, because the application reads data in bursts. When the packet buffer size is too small, the decoder can not decode the video frames in time, leading to jerky playback and video frames that are displayed too late.

For fluent playback of the IPTV stream, the prototype IPTV client needs to have buffered at least 500 KB of data before playback is started. Therefore when determining the minimum buffer requirements for enabling retransmissions, one possible lower bound is already known: the buffering requirements for decoding and displaying the IPTV stream.

Experiment settings

For the experiment the following settings are used:

Downlink delay:	10 ms
Downlink loss:	5% (uncorrelated loss)
Uplink delay:	2 ms
Uplink loss:	0%
Client buffer size:	260 - 1300 KB
IPTV stream:	3.6 Mbit/s
IPTV stream IP packet size:	1356 (fixed)
IPTV retransmission packet size:	1358 bytes
Retransmission Cache buffer size:	78 KB
Experiment duration:	120s

Results

In figure 5.8 the cache hit ratio, the retransmission success ration and the delayed frames ratio are plotted as a function of the buffer size for the experiment with at most one retransmission attempt per packet. Immediately it is clear that the buffering requirements for the application to operate without buffer under runs, are higher than the buffer requirements for packet retransmissions. When the buffer fill level is ranged between 230 and 500 KB, a high percentage of the displayed video frames is displayed after the scheduled presentation time. The reason for this is that the decoder has to wait for all data of a certain video frame to be available. When there is not enough data for the next frame to be decoded (i.e. not all RTP packets containing data from this frame are received), the decoder has to wait until this data is available. This delay thus leads to jerky playback.

Although the IPTV client is not able to provide fluent playback, the packet retransmission mechanism does work: up to 95% of all missing packets are recovered. One can however argue whether these packets are really recovered in time, because of the playback problems.

When the buffer fill level is set to 500 KB or higher, the IPTV client can provide fluent playback. Any further increase of the buffer does not lead to a significant improvement of the number of packets that can be recovered: the ratio stays between 94.5 and 95.5%. When the buffer is set to 520 KB, the startup delay (the time between starting the application and displaying the first frame on a screen) is approximately 1.1 seconds.

What can be concluded is that the minimum required buffer size for the IPTV client is determined by the buffer requirements for the decoding and displaying of the IPTV video stream, not by the requirements for packet retransmission.

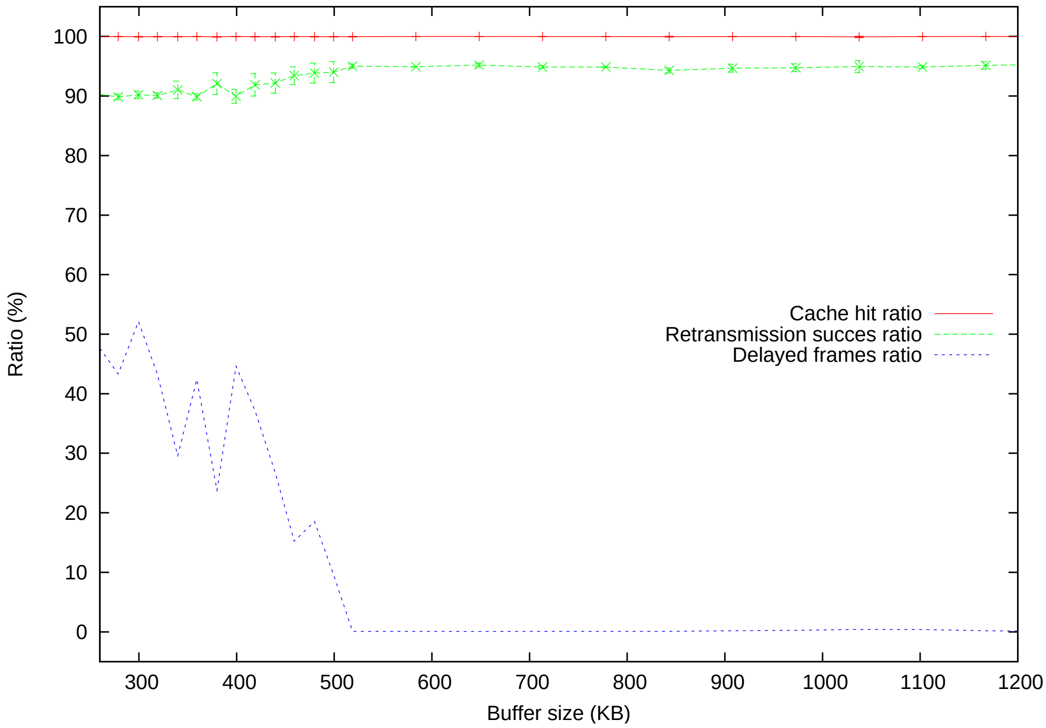

Figure 5.8: Buffer dimensioning IPTV Client, one retransmission attempt

Figure 5.9 shows the the results for the experiment with up to three retransmission requests per packet. Again the results show what is already expected: the minimum size size of the buffer is determined by the buffer requirements for decoding and displaying. A buffer fill level size of 520 KB is sufficient to allow multiple retransmission attempts per packet: between 99.3 and 100% of all missing packets are recovered before they are required for decoding.

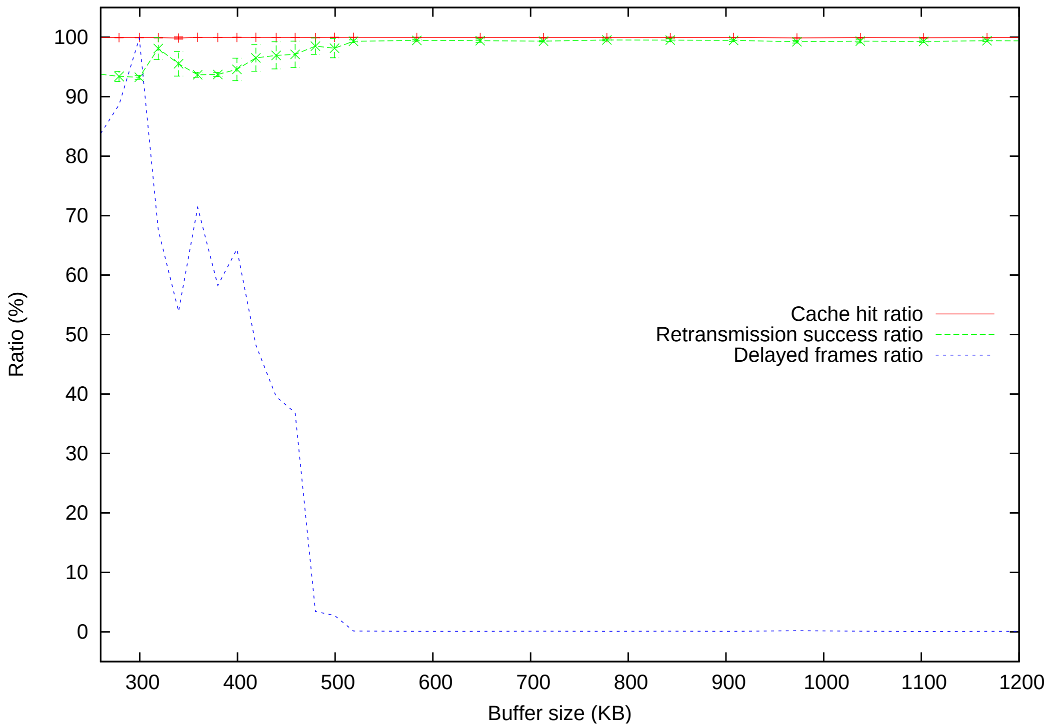

Figure 5.9: Buffer dimensioning IPTV Client with three retransmission attempts

5.4.3 Conclusions

The experiment results show what the minimum required buffer size of the Retransmission Cache must be for an access network with a downlink propagation delay of 10 ms and a uplink propagation delay of 2 ms. When one retransmission attempts is allowed per packet the buffer size needs to be at least 40 KB, when multiple attempts are allowed, the buffer size increases to 60KB.

The IPTV client buffer dimension experiment results show us that the applicability of packet retransmission is not determined by the time needed to allow for retransmissions, but by the minimum buffer size requirements of the IPTV application.

The experiment was performed with a IPTV stream with a throughput of 3.6 Mbit/s. For other IPTV stream configurations (with a higher or lower bit rate) however, the same behavior can be expected. When the bit stream size increases, the size of the encoded frames will also increase. This means that the size of the IPTV client buffer and the size of the Retransmission Cache need to increase proportionally. For the IPTV client the buffer must increase to be able to fit the largest encoded frame, for the retransmission cache the buffer must be increased to support the additional packets that are sent in the same time interval, as the in this experiment presented results. The same argument also holds for alternative downlink and uplink propagation delays. Validation of these assumptions is not provided in this thesis, so this is left for future work.

5.5 Packet loss recovery in an uncongested network

The second experiment investigates the applicability of the packet retransmission mechanism under different loss scenarios in an uncongested access network. First the mechanism is tested in a network setup with uncorrelated packet loss; the second part of the experiment investigated the performance in a network with correlated (burst) packet loss.

The results are provided in graphs which show the perceived loss rate after applying packet retransmissions as a function of the introduced packet loss rate. Furthermore the network overhead for packet retransmissions is investigated. For this experiment the same network setup is used as for the buffer dimensioning experiments, as can be seen in figure 5.10.

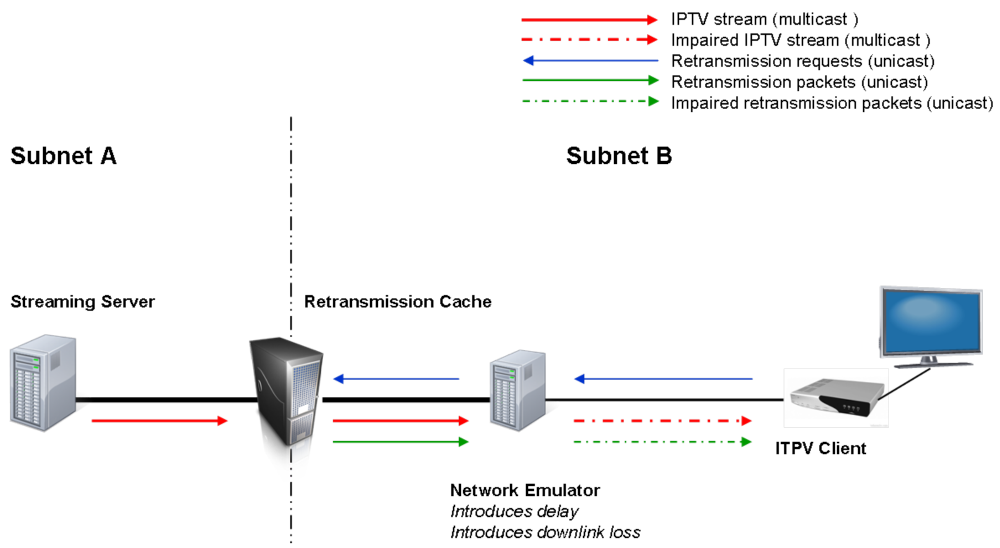

Figure 5.10: Test setup network - Uncongested Access Link

5.5.1 Uncorrelated packet loss

The first part of the experiment focuses on the applicability of packet retransmissions when packet losses are uncorrelated. The IPTV client receives and decodes the IPTV stream for 60 seconds. During this interval the network emulator randomly drops packets in the downstream direction. This affects packets from the IPTV stream, as well as packets being retransmitted. The experiment is performed for loss rates ranging from zero to ten percent.

Experiment settings

For the experiment the following settings are used:

Downlink delay:	10 ms
Downlink loss:	0-10% (uncorrelated loss)
Uplink delay:	2 ms
Uplink loss:	0%
Client buffer size:	790 KB
IPTV stream:	3.6 Mbit/s
IPTV stream IP packet size:	1356 (fixed)
IPTV retransmission packet size:	1358 bytes
Retransmission Cache buffer size:	98 KB
Experiment duration:	60s

Results

In figure 5.11 the perceived loss after applying retransmission is plotted as a function of the introduced introduced packet loss rate. This is provided for one (R=1), two (R=2), or three (R=3) allowed retransmission attempts per packet. The figure first of all shows that when packet retransmission is enabled the retransmission mechanism reduces the packet loss rate for the application significantly, to below 1.2% when 10% packet loss is introduced. When the amount of allowed retransmissions is increased to two (R=2), the packet loss rate can be further reduced to below 0.4%.

The graph also shows that allowing allowing a third retransmission attempt (R=3)does not bring any further (significant) recovery of missing packets. This can be explained by the delay between subsequent requests (the interval before a retransmission request expires). The gain of allowing more retransmissions is limited to the size of the buffers and thus restricted by application delay requirements (the maximum allowed startup delay).

In the worst case (10% loss), the retransmission mechanism was able to recover 96.6% of the missing packets with at most two retransmission attempt. When the network loss is at most 5%, the retransmission mechanism recovers 99% of the missing packets with at most two retransmission attempt.

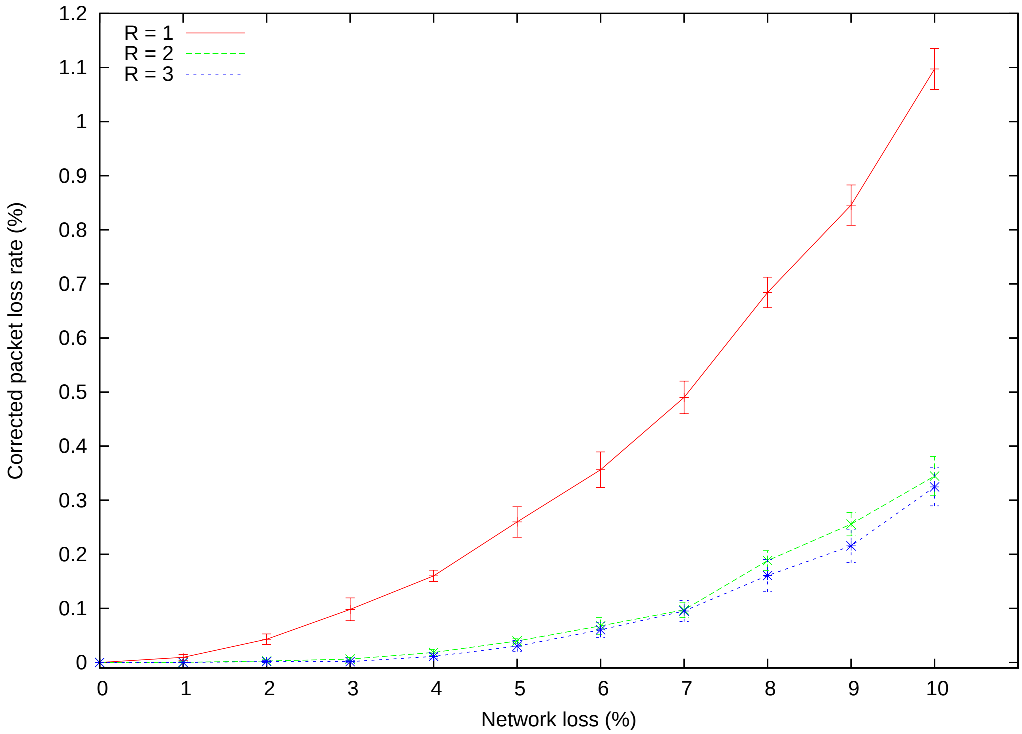

Figure 5.11: Error recovery performance for uncorrelated packet loss for one (R=1), two (R=2) or three (R=3) allowed retransmission attempts

In figure 5.12 the retransmission bandwidth requirements are shown, as a percentage of the bandwidth needed for the IPTV stream. What the graph clearly shows is that the bandwidth needed for the retransmission stream is small compared to the bandwidth needed for the IPTV stream and proportional to the introduced packet loss rate. The bandwidth requirements will therefore not likely be a problem when the mechanism is implemented in a live access network environment.

The distribution of the number of retransmission requests per RTCP packet is presented in figure 5.13. As what was expected the feedback messages contain only a lower number of request; the majority contains only one request, while a minor part contains two requests. When the packet loss is random (without correlation), the BLP component of the Generic Nack feedback message does not provide a real benefit. A reduction of the number of transmitted feedback messages would require the delay of the the transmission of the feedback packets, such that additional requests can be added to the Generic Nack message after it has been created. This would require an evaluation of allowed delay before transmitting a retransmission request.

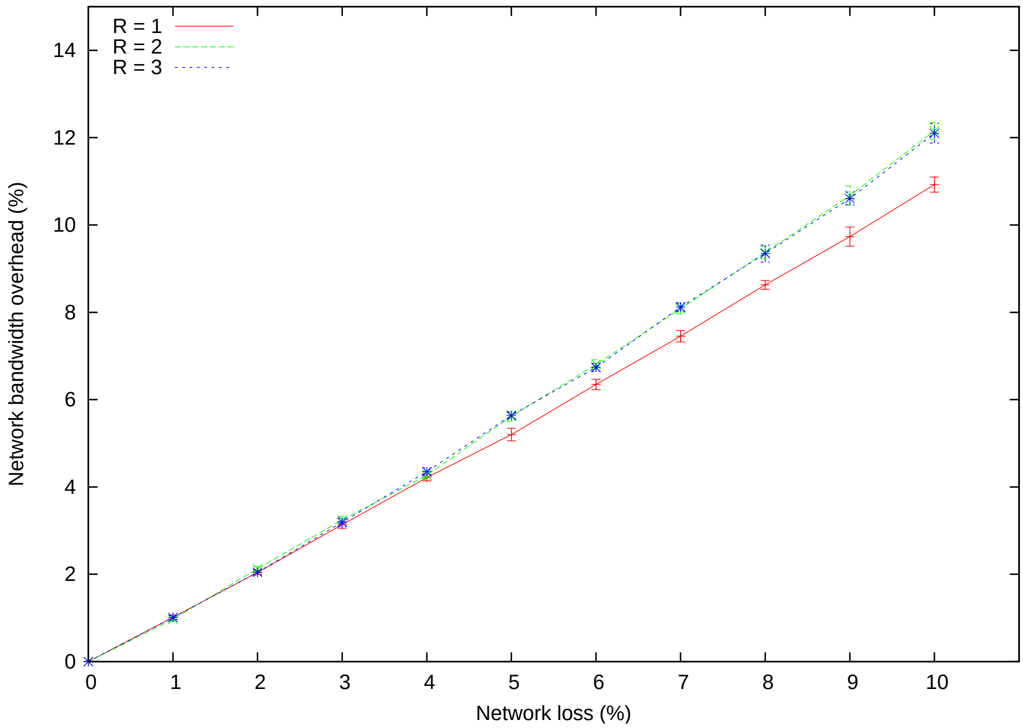

Figure 5.12: Retransmission Bandwidth overhead for uncorrelated packet loss for one (R=1), two (R=2) or three (R=3) allowed retransmission attempts

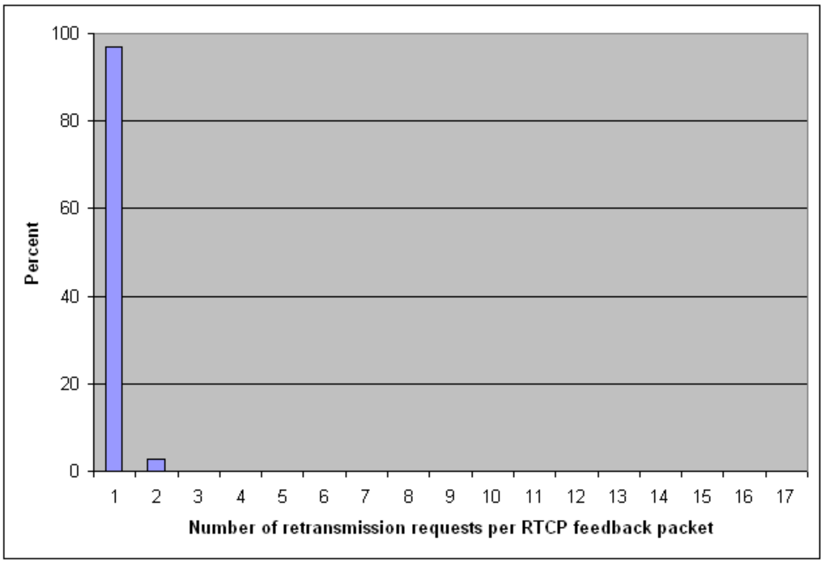

Figure 5.13: Feedback message size distribution for uncorrelated packet loss

5.5.2 Correlated packet loss

For the evaluation of the packet retransmission mechanism in a network environment where correlated (burst) losses occur, several trace files are created which contain the results of a Gilbert-model. The average introduced loss rates are identical to the values used in the previous experiment, but now the packet losses are correlated. For creating burst losses a Gilbert model with the following settings is used:

1. The Gilbert model is a Markov model with two states
   (a) State ‘G‘ presents a good state, in which packet loss has a low probability.
   (b) State ‘B‘ presents a bad state in which the changes for subsequent losses have a high probability.
2. A transition from G to B presents the loss of a packet, when the previous packet was transmitted successfully.
3. A transition from G to G presents the consecutive successful transmission of a packet.
4. A transition from B to B presents the consecutive loss of a consecutive packet.
5. A transition from B to G presents the successful transmission of a packet after one or more consecutive packets have been lost.
6. The state transition probabilities are:
   (a) State transition BB occurs with probability $4/5$;
   (b) State transition BG occurs with probability $BG = 1 - BB = 1/5$.
   (c) State transition GB occurs with probability $ER * BG/(1 - ER)$, where ER is defined as the average loss rate.
   (d) State transition GG occurs with probability $1 - GB$.

With these settings ten trace files of 200000 transitions are created with the numerical computation tool GNU Octave [76]. The trace files are then used in the network emulator to influence the IPTV stream and the retransmission packets.

Experiment Settings

For the experiment the following settings are used:

Downlink delay:	10 ms
Downlink loss:	0-10% (correlated loss)
Uplink delay:	2 ms
Uplink loss:	0%
Client buffer size:	790 KB
IPTV stream:	3.6 Mbit/s
IPTV stream IP packet size:	1356 (fixed)
IPTV retransmission packet size:	1358 bytes
Retransmission Cache buffer size:	98 KB
Experiment duration:	120s

Results

In figure 5.14 the results are shown for the burst loss emulation experiment. When these results are compared with the results for uncorrelated packet loss, one difference is noticeable: the performance of the retransmission mechanism is lower with correlated packet loss. When 10% network loss is introduced, the corrected packet loss rate for one allowed retransmission attempt lies between 2.13% and 2.23%. So now the IPTV client is only capable of recovering 78% of all missing packets (for uncorrelated loss this percentage was 88%).

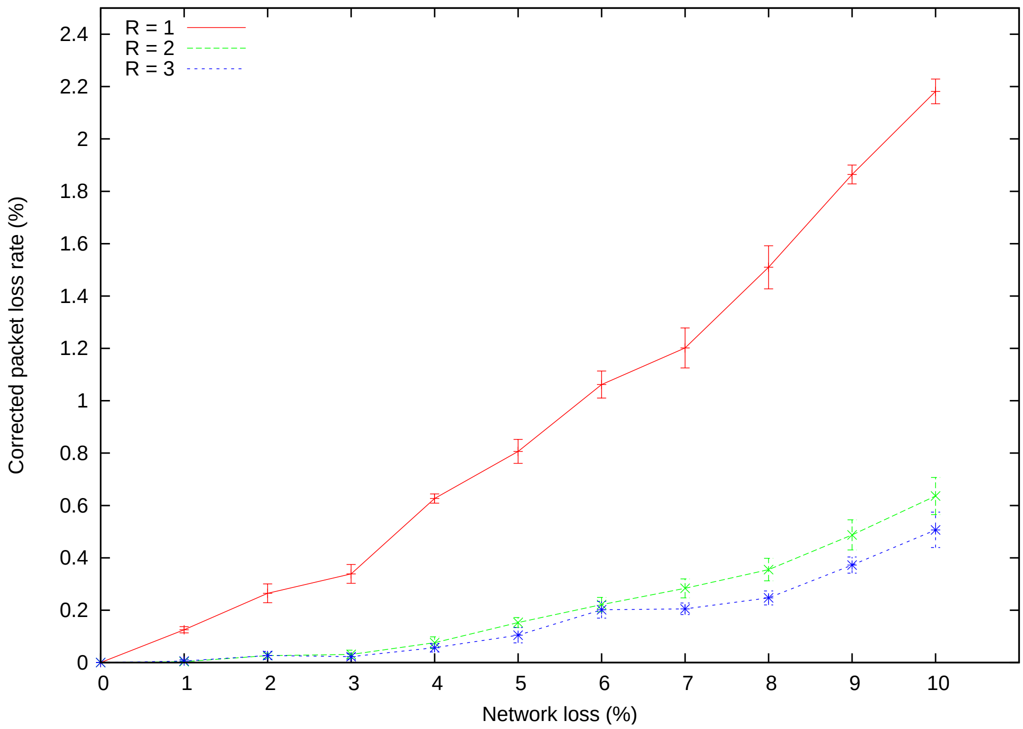

Figure 5.14: Error recovery performance for correlated packet loss for one (R=1), two (R=2) or three (R=3) allowed retransmission attempts

Two possible reasons for this difference are:

- The packet loss detection mechanism. Because the detection of packet loss depends on the arrival of a subsequent packet, when burst packet loss occurs, the loss will be detected later then when the losses are isolated.
- The likeliness of retransmission packets being caught in a burst. Due to the bursty nature of the packet loss, the distance between retransmission packets is smaller: multiple requests are sent in one feedback message, which result in multiple subsequent retransmission packets. These retransmission packets may be dropped in a burst.

When a seconds retransmission attempt is allowed, the corrected packet loss ratio is further reduced to below 0.71% for a introduces packet loss rate of 10%; when the packet loss rate is 5% or smaller the corrected packet loss rate is at most 0.17%. As what can be expected the third only offers minor improvements compared to two allowed retransmission attempts.

In figure 5.15 the network bandwidth overhead for packet retransmissions is plotted as a function of the introduced network lossed. The figure shows similar results as what was presented in the previous experiment: the bandwidth needed for the retransmission stream is proportional to the introduced packet loss rate.

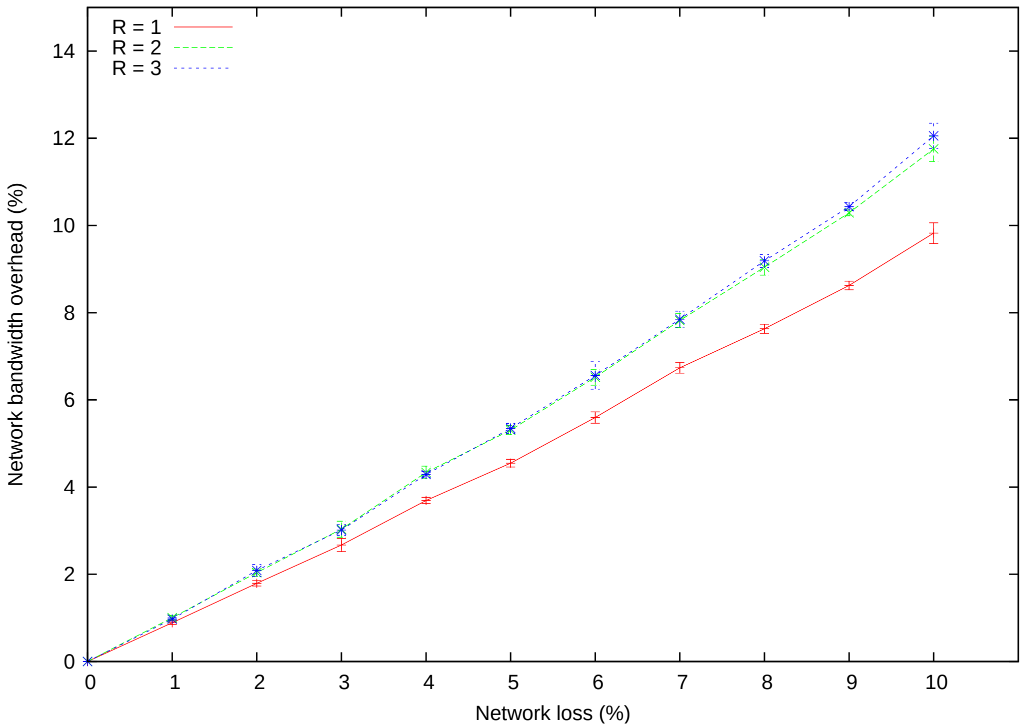

Figure 5.15: Retransmission bandwidth, burst network loss

For the evaluation with uncorrelated packet losses the Generic Nack message size distribution showed that most messages contained only one retransmission request. In figure 5.16 the message size distribution is shown for correlated packet loss. In this case only 18% of the feedback messages contain only one retransmission request. And all other possibilities, ranging from 2 to 17 retransmission requests per packet, are also in use, so for burst traffic loss the feedback messages are used more efficiently. What might seem strange is the increase from 16 to 17 requests per retransmission packet. This is not caused by an increase of burst losses of exactly 17 consecutive packets, but by the fact that all loss sequences of more then 17 consecutive packets are split, as the Generic Nack message can contain at most 17 retransmission requests.

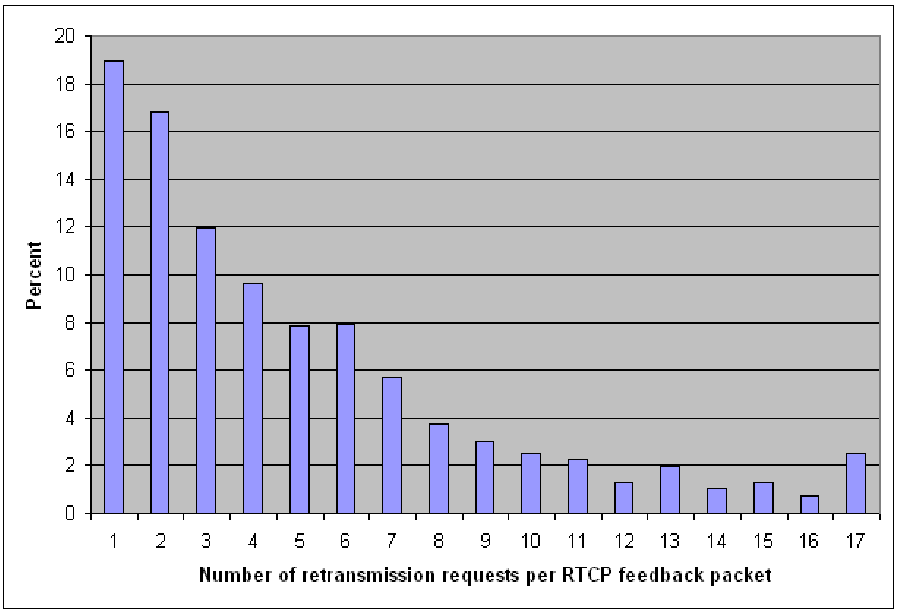

Figure 5.16: Feedback message size distribution for correlated loss

5.5.3 Refined retransmission timeout

Both the results for uncorrelated and correlated packet loss sthowed showed only minor packet recovery improvements when the number of allowed retransmission attempts is increased from two to three. To determine if this improvement depends on the client buffer size or on the retransmission time-out interval, the retransmission timeout function is reconfigured such that the timeout interval is reduced:

$$RTO_{n+1} = 2 * RTT + 1.5 * rttvar_{n+1} \tag{5.2}$$

is changed into:

$$RTO_{n+1} = 1.4 * RTT + 1.5 * rttvar_{n+1} \tag{5.3}$$

This reduction can have two consequences:

- The time between the first and second retransmission attempt and the time between the second and the third retransmission attempts are reduced, which means there is more time available for a succesfull recovery with a third retransmission attempt.
- Due to the reduction in timeout interval, retransmission attempts may be considered unsuccessful too early; a new retransmission request is made while the retransmission packet for a previous attempt is send, but not yet received by the IPTV client. This can lead to unnecessary duplicate

retransmissions, which is a waste of bandwidth.

Figure 5.17 shows the results for the experiment with the refined timeout mechanism. Although the results for one or two allow ed retransmission attempts do not differ from previos results, the refined timeout mechanism seems to improve the packet recovery performance of the retransmission mechanism when three retransmission attempts are allowed, as thse resulting packet loss rates are lower then the loss rates achieved with the original settings.

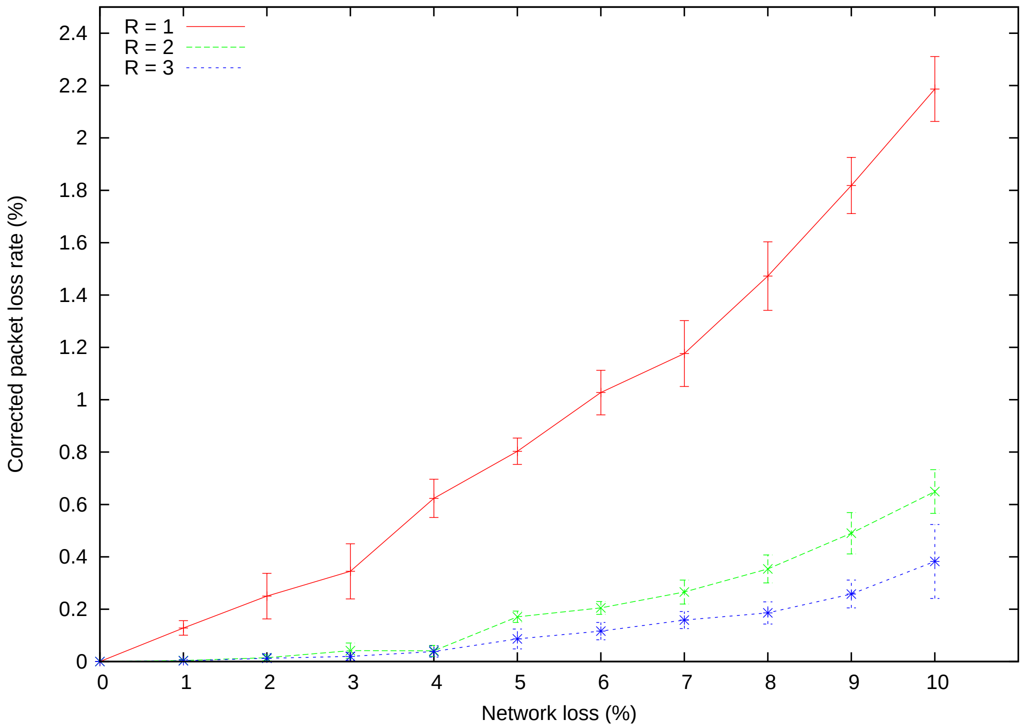

Figure 5.17: Error recovery performance for correlated packet loss with a refined retransmission timeout mechanism

Figure 5.18 shows a comparison of the percentage of duplicate retransnsisions for the original and refined timeout mechanism for two (R=2) and three (R=3) allowed retransmission attemppts. The improvement of the packet loss ratio apparently comes at a cost: due to the refined time-out mechanism between 0.75 and almost 2% of all received retransmission packets are duplicates.

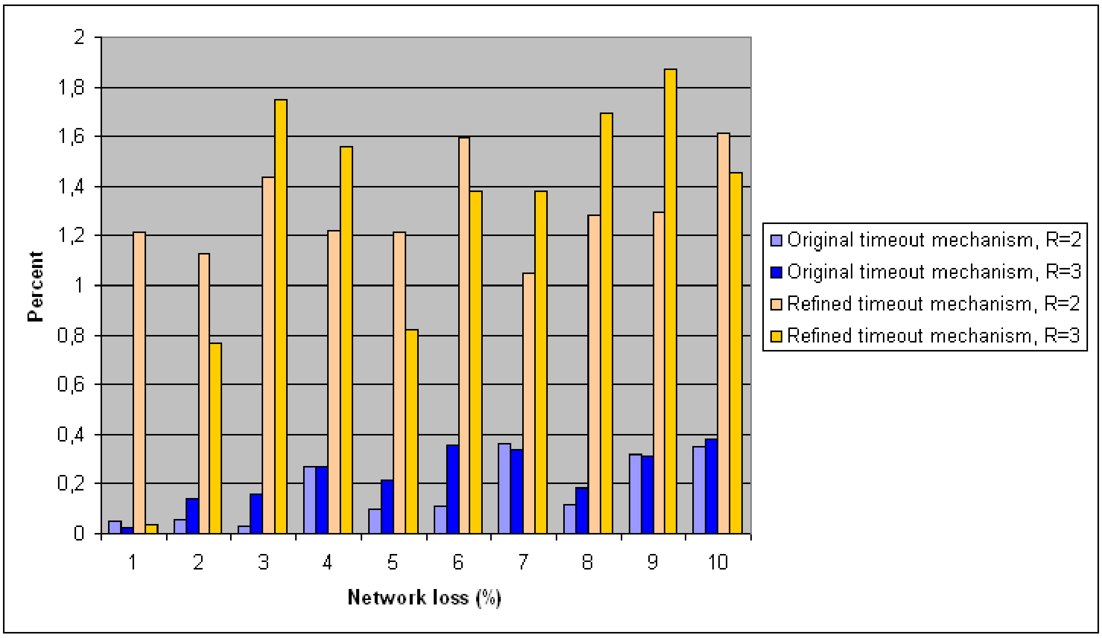

Figure 5.18: Duplicate packet retransmissions for the original and refined retransmission timeout mechanisms

5.6 Packet loss recovery in a congested network

The final experiment focuses on investigation the effects of the retransmission mechanism in a network environment where packet loss is caused by congestion. For this purpose the access link capacity is limited to 6 MBit/s and additional traffic is generated to create congestion. The additional traffic is a TCP-traffic stream, which represents a simulated large FTP file transfer.

Because a packet retransmission mechanism generates additional traffic when packet loss occurs, the mechanism itself can become a drawback when packet loss occurs by congestion. To prevent this problem, the TCP protocol reduces the transmission rate. With other words, the TCP control is aware of congestion and tries to prevent it. A congestion prevention mechanism is possible with RTP: a sender can reduce it's transfer rate when a client reports a lot of packet loss. Furthermore do RTCP transmission constraints prevent overloading the network with RTCP packets.

For linear broadcast TV, the transmission bandwidth will not be reduced when congestion occurs (as this would reduce the IPTV video quality for all users). Furthermore does the prototype implementation not adhere to RTCP transmission constraints.

The transmission of additional packets, the retransmission packets, may lead to worse congestion conditions because the additional bandwidth requirements may not be available. This may even lead to worse IPTV packet loss rates, as more packets of the IPTV stream would get dropped.

This experiments investigates what happens when congestion is the cause of packet loss: if the uncorrected packet loss rates increases due to retransmissions and if the packet retransmissions can still lead to a (perceived) reduced packet loss rate.

Experiment settings

In figure 5.19 the network configuration for the experiment is presented. The network emulator limits the down link throughput to 6 Mbit/s and starts dropping packets when the bandwidth limit is reached. The 6 Mbit/s capacity is sufficient to transmit the IPTV stream without dropping packets.

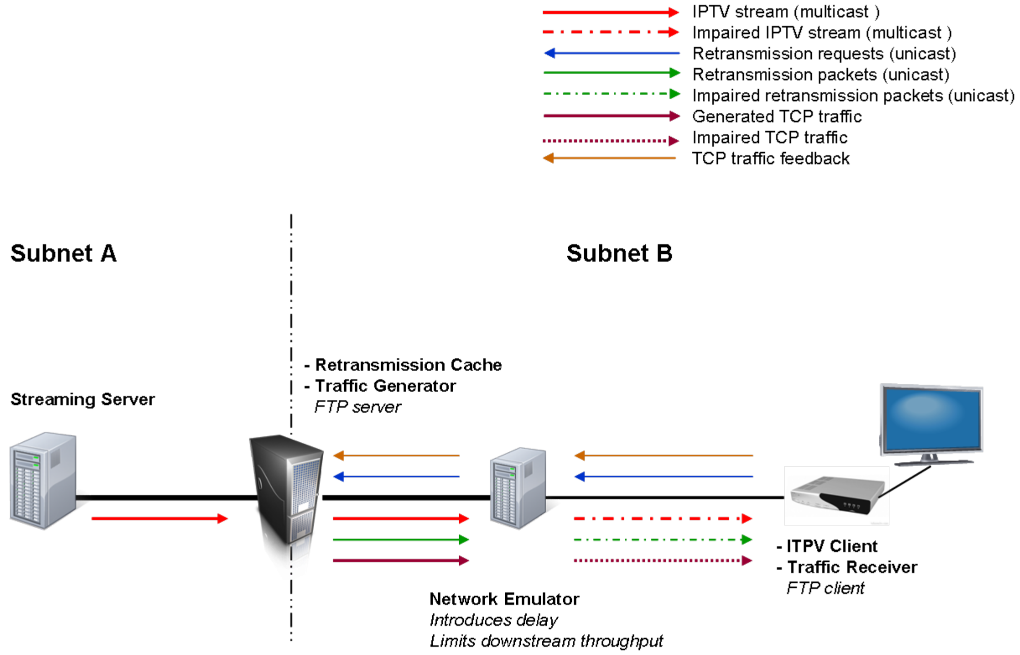

Figure 5.19: Congested network setup

In the Retransmission Cache system a traffic generator is placed, which will send TCP traffic to the IPTV client for 30 seconds, emulating a FTP file transfer. The IPTV client therefore is equipped with a traffic receiver, resembling a FTP client. During the experiment the IPTV client will receive the IPTV stream for 60 seconds. After 10 seconds the TCP traffic stream is started. The TCP stream transmission will result in congestion in the network emulator.

Due to the congestion both the IPTV stream and TCP stream will be effected. The TCP transfer will, due to occurring packet loss, adapt its transmission. The IPTV transfer will not adapt, as the Streaming Server is not aware of the losses (besides the fact that RTP on top of UDP does not provide congestion control). The packet loss only leads to packet retransmission requests and retransmissions, if the retransmission functionality is enabled in the IPTV client. Retransmission packets may however also get dropped.

Besides gathering statistics in the Retransmission Cache and the IPTV client, the data sent between the network emulator and IPTV client is captured with the network capturing tool Wireshark[77]. The captures are used to analyze the IPTV, retransmission and TCP stream.

The experiment is first conducted with the retransmission mechanism disabled. This provides insight in the effects of congestion to the delivery of the IPTV stream and the file transfer. It also provides the uncorrected packet loss rate for the IPTV stream. Then the experiment is repeated, but with packet retransmission enabled. The results of this experiment run, consisting of the uncorrected packet loss rat and the corrected packet loss rate, can then be compared with the results of the run without packet retransmission.

For the experiment the following settings are used:

Downlink delay:	10 ms
Downlink loss:	0%
Uplink delay:	2 ms
Uplink loss:	0%
Client buffer size:	780 KB
IPTV stream:	3.6 Mbit/s
IPTV stream IP packet size:	1356 (fixed)
IPTV retransmission packet size:	1358 bytes
Retransmission Cache buffer size:	98 KB
Experiment duration:	60s
Access link throughput:	6 Mbit/s
TCP file transfer duration:	30s

Results

Figures 5.20 and 5.21 show the IPTV stream, the TCP stream and the retransmission packet stream flows and the total throughput of the access link. Figure 5.20 shows the results for the evaluation without packet retransmissions, the 5.21 shows the results with packet retransmissions enabled. Both results show how the TCP flow adapts to the available bandwidth, by using all available bandwidth that is not used for the IPTV stream and the eventual retransmission packets. Because the round trip times between the TCP sender and receiver are quite low (circa 12 ms), TCP can rapidly adapt to the available bandwidth. Figure 5.21 shows that the congestion will lead to packet retransmissions. What is interesting to determine, is whether the packet retransmissions lead to additional losses of IPTV stream packets, or that the additional packets will only effect the TCP flow.

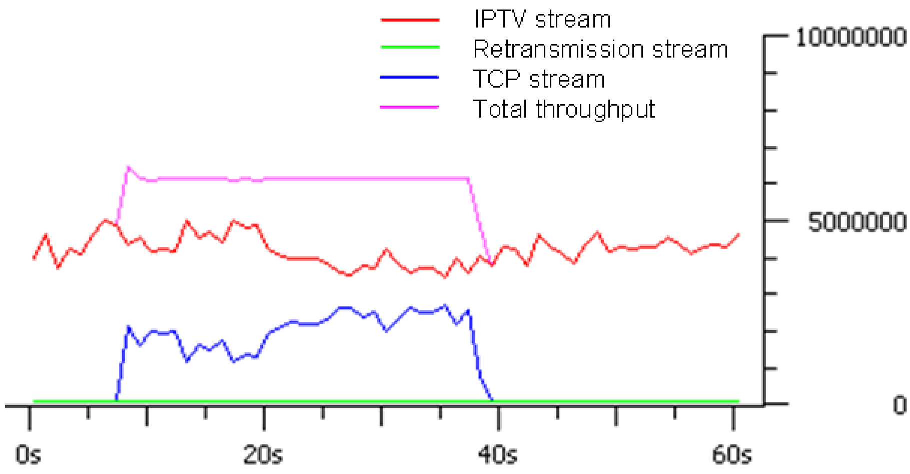

Figure 5.20: Congestion trace - without retransmissions

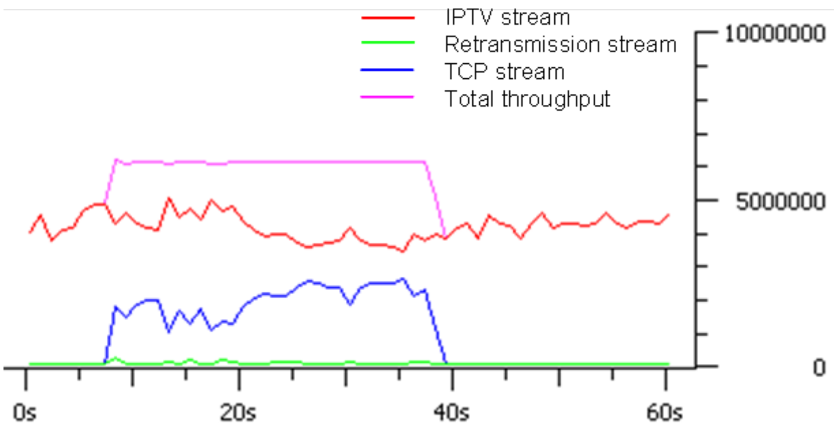

Figure 5.21: Congestion trace - with retransmissions

Figure 5.22 shows a comparison of the uncorrected and corrected packet loss ratios for both experiments, based on 10 measurements each.

As expected the uncorrected packet loss rate equals the corrected packet loss rate for the experiment where packet retransmissions were disabled. When packet retransmission is disabled, the corrected packet loss rate is reduced significantly, but a more important observation can be made: the uncorrected packet loss rate is significantly higher when retransmission is enabled then when retransmission is not offered. The loss ratio increases from 0.85 to 1.1% perrcent. This means that the packet retransmissions negatively affected the delivery of the IPTV stream, but due to the retransmission mechanism, the variable transfer rate of the IPTV stream and the adaption of the TCP stream to the congestion, the corrected packet loss ratio is still lower then in the case that the retransmission mechanism is disabled.

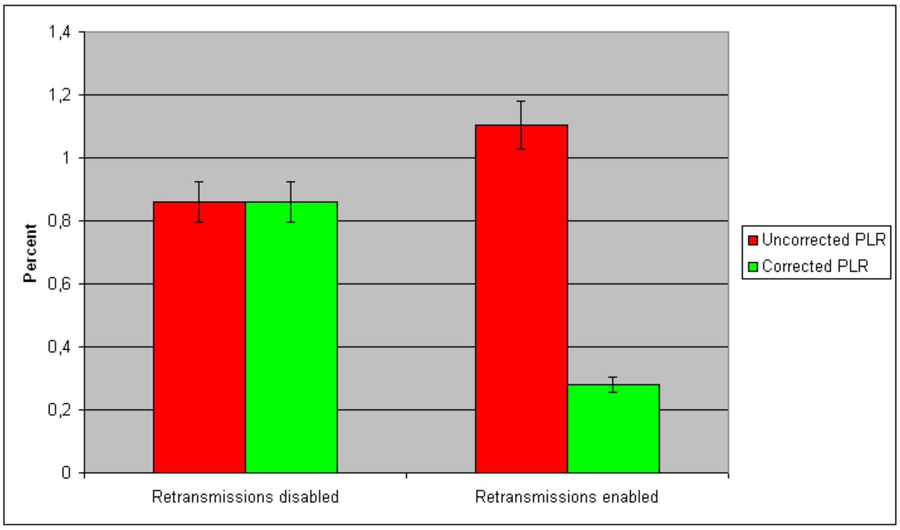

Figure 5.22: Retransmission in congested network

5.7 Applicability and scalability

The previous experiment results showed that packet retransmissions for broadcast TV that is distributed using multicast IP networks leads to a large reduction of the perceived packet loss rate. These experiments however were performed with only one IPTV client, while in practice up to a couple of thousand of IPTV clients may be connected to one access node. Scalability experiments however could not be performed, due to two reasons: first there was a lack of time to perform additional experiments. Secondly, the scalability experiments can not easily be performed in a test network setup, which from a practical point of view cannot support a lot of IPTV clients (it would for instance require a lot of require a lot of systems that run IPTV clients). The evaluation of the scalability of the retransmission mechanism can therefor more easily be performed in a (network) simulator, but this is left for future work. However some concerns about the scalability and applicability of the retransmission mechanism are provided in the following sections.

5.7.1 IPTV client buffer requirements

The IPTV client buffer dimensioning experiments showed that the required buffer size depends largely on the application buffer requirements with an upper bound to the application buffer requirements plus the buffer size of Retransmission Cache. IPTV Set Top Boxes currently have at least 64-128 MB of memory, so the memory resources required for packet retransmissions should not be a concern. The buffer size of the Retransmission Cache in the experiment was only 10% of the buffer space required by the application, so the delay imposed by enabling packet retransmissions can be considered acceptable.

5.7.2 Retransmission Cache buffer requirements

When looking at the buffer requirements for the retransmission cache to allow for successful application of the retransmission mechanism, the results showed that a buffer size ranging from 25 KB to 80 KB is required, which can be considered moderate for an access node device. This buffer size will not increase when more IPTV users will be using the retransmission mechanism, as all users that are watching a specific TV channel can use data from the same retransmission cache. Per IPTV channel for which retransmission functionality would be required an additional Retransmission Cache would be needed, therefore the total memory requirements depend on the number of television channels that require retransmission functionality. Because the retransmission mechanism in principle can be enabled dynamically, a service provider could adapt the memory requirements based on loss characteristics and popularity of a TV channel: if a specific threshold is reached for the number of simultaneous viewers of a channel, or a threshold regarding the loss reported by IPTV clients, caching for that client can be enabled to allow retransmissions; when these thresholds are not reached, usage could be disabled, therefore saving memory resources.

5.7.3 Network and processing requirements

As discussed in section 4.2.3 packet retransmissions for multicast IPTV delivery can be provided using multicast or unicast delivery. The implemented prototype uses unicast delivery, because of the lower complexity and because this does not lead to an increased load on all access links, when only one client asks for packet retransmission: unicast retransmissions will only be sent to the IPTV client asking for retransmissions, so only his access node will be affected by additional traffic.

This means that when the number of IPTV users requesting retransmissions increases, the Retransmission Cache needs to process more retransmission requests and needs to transmit more packets. These transmissions may contain a lot of duplicates because every IPTV client needs to request it's own retransmission. Therefore two bottlenecks can be identified: the processing power of the Retransmission Cache and the packet processing capabilities of the Access Node. To determine how many users can be serviced by one Retransmission Cache additional experiments are required, which focus on the scalability of the retransmission mechanism.

Chapter 6

Conclusions and future work

This chapter discusses the conclusions and the future work. First the research questions are answered. Then the overall conlusions are given and discussed. Finally some recommendations for future work are given.

6.1 Results

This section answers the research questions which are described in section 1.3.

1. What are the effects of packet loss on IPTV streaming applications?

When a IPTV stream suffers from packet loss, the Set Top Box is not able to decode the video stream correctly because of the data that is missing. This will result in the impairment of the decoded video frames, which is noticeable as visual errors, wrongly decoded blocks and image distortions.

The severity of the impairments depend on a number of factors. Due to the nature of video compression techniques, visual errors can propagate in subsequent frames. When a reference frame is impaired, all subsequent frames until the next reference frames will also be impaired. The duration of the error therefore depends on the type of frame that is impaired and whether other frames are referencing the frame that is impaired.

2. What techniques can be used to provide error resiliency for IPTV streaming applications?

A reduction of the noticeable effects of packet loss can be provided by means of payload interleaving, error concealment functionality of the video codec or by giving priority for specific application payload data, such as reference frames. Bandwidth adaptation can be used to prevent or reduce congestion, but is not very practical for multicast stream delivery. Packet loss recovery can be provided by means of Forward Error Correction or packet retransmissions. Both techniques have their limitations for multicast delivery as mentioned in 2.14. An alternative would be to offer packet retransmissions in a subtree of an multicast distribution network. This allows for the rapid recovery of missing packets and avoids perforances isusues related to feedback implosion when retransmissions are provided between the Streaming Server and IPTV clients.

3. How can fast retransmissions be provided for multicast IPTV stream delivery service using the Real-time Transport Protocol?

The RTP protocol does not provide the possibility of sending time-constrained feedback or a feedback message type that can be used to indicate the loss of individual packets. Both of thse are requirements for packet retransmission mechanism. These issues have been addressed with a protocol extension that offers a new mechanism to provide immediate feedback and specifies a generic feedback RTCP format that can be used to notify the loss of specific packets (RFC 4585). Another recent RTP extension provides a RTP retransmission payload format, which can be used to provide retransmission of RTP packets, regardless of the original payload (RFC 4588). These extensions thus provide the building blocks for allowing the RTP protocol to be used for a packet retransmission mechanism.

In the proposed design packet retransmissions are offered by a Retransmission Cache in an access node. This has three main advantages. Because the packet recovery is offered in only a subtree of the multicast distribution path, feedback implosion is not likely to occur. Furthermore is the packet recovery offered in those parts of the network were the problems occur. Finally, because the retransmission functionality is offered in the access node, packet retransmissions are very fast, which is essential for packet recovery for a time-constrained IPTV service.

The Retransmission Cache temporally caches the RTP packets transmitted to the IPTV client. When an IPTV client misses packets, it will ask for retransmissions using the feedback format described in RFC 4585. The Retransmission Cache will retrieve the packet from the cache and transmit the packet to the IPTV client using the retransmission payload format. By allowing multiple retransmission attempts per packet an IPTV client can improve the effectiveness of the retransmission mechanism.

4. How can the effects of error resiliency based on packet retransmission be measured?

Chapter 2 discussed several metrics and measurement techniques that can be used to evaluate video quality. A lot of measurements methods cannot be used in a live IPTV setup, either because the measurements require a reference video signal, or the measurements require interaction with the user. Alternatively, network metrics can be used to evaluate the effect of error resiliency based on packet retransmission.

For the evaluation of the effectiveness of the packet retransmission mechanism two network layer metrics are used: the uncorrected packet loss rate and the corrected packet loss rate. The former describes the percentage of packets that are considered lost before error recovery is provided by means of packet retransmissions. The latter describes the percentage of packets that are still lost after error recovery by means of packet retransmissions. The difference between this metrics describes the effectiveness of the packet retransmission mechanism.

5. What are the parameters that influence the performance of the RTP retransmission mechanism?

The performance experiments have revealed several parameters that (significantly) influence the retransmission mechanism's performance. First of all, the buffer sizes are an important aspect to make packet retransmissions feasible. The experiment results show what the buffer size of the Retransmission Cache must be to allow for the successfully recovery of a packet by means of one or multiple retransmission attempts. The lower bound for the required buffer size for the IPTV client is determined by the playout buffer requirements for the decoding and displaying of the IPTV stream. The additional buffer size requirements for retransmissions are upper bounded by the buffer size of the Retransmission Cache, but since the play-out buffer can successfully be used for applying retransmissions, the upper bound may even be smaller. The IPTV service delay requirements ultimately determine the the size of the buffer and therefore the amount of time available for successful packet recovery. Therefore there is a trade-off between the performance of the retransmission mechanism and the application startup delay.

Two other important parameters influencing the performance of the retransmission mechanism are the number of allowed retransmission attempts per packet in combination combined with the expiration of retransmission requests. The experiment results show a trade-off between an improved recovery rate when the retransmission time-out is decreased and an increase in the number of duplicate retransmissions.

A final parameter that influences the performance is the packet loss detection mechanism. In the prototpe packet loss is detected by means of gaps in sequence numbers. By also enabling a timeout mechanism for the next expected packet the recovery mechanism can be improved when burst losses occur.

6. For which network conditions can RTP-based packet retransmission be successfully applied as an error resiliency mechanism?

The results of the experiments discussed in chapter 5 show that even for a time-constrained service like broadcast IPTV the proposed retransmission mechanism leads to significant reduction of the packet loss rates. The results show that the retransmission mechanism effectively reduces loss in networks with uncorrelated losses ranging from 1 to 10%, while in a network with correlated losses the mechanism is slightly less effective.

In a congested network a reduction in the corrected packet loss rate was achieved, even when the retransmissions initially lead to a higher uncorrected packet loss rate, i.e. contributed to congestion. That the retransmission mechanism lead to a performance increase is explained by the following two factors. First, in the experiment a congestion aware file transfer was used. A TCP transfer adapts to the available bandwidth and reduces the transfer rate when congestion occurs. This allowed more IPTV stream packets and retransmission packets to pass the network without problems. Second, because of the variability of the IPTV stream packet losses that occur during the peaks of the IPTV stream can be recovered by retransmissions when the IPTV uses less network resources.

6.2 Conclusions

This master thesis project studied and investigated how packet loss affects IPTV services and how error resiliency for multicast IPTV broadcast television can be provided. This started with a background study investigating the IPTV domain and relevant techniques and technologies. Based on the acquired background information, the literature study performed at NEC and the results of the IPTV function study performed at NEC the requirements for a fast retransmission function for multicast IPTV broadcasts were proposed.

A system design was proposed compliant to this fast retransmission function. This system uses RTP streaming and feedback functionality provided by the Real-time Control Protocol for delivery of an IPTV stream and retransmissions of packets that are considered lost by an IPTV client. In this design packet retransmissions are offered by a Retransmission Cache placed in an access node, which allows for a fast recovery of packets indicated lost by an IPTV client. The packet retransmissions are provided using a different RTP session allowing the retransmission functionality to be dynamically enabled, and because the mechanism uses a new component (the Retransmission Cache), the functionality can be inserted in a existing multicast IPTV network infrastructure, without affecting the already available IPTV stream delivery.

This design was implemented in a prototype, which functions as a proof an concept and was used for evaluation of the retransmission functionality by means of experiments in a test network setup. The goals of these experiments were to figure out under which network conditions retransmissions could be applied successfully and to figure out which parameters influence the effectiveness of the retransmission mechanism.

The experiment results show that offering packet retransmissions for multicast IPTV broadcasts is feasible and can lead to significant reduction of the packet loss rates under various packet loss conditions in the access network. The results also show that the memory requirements for enabling packet retransmissions in an access node are moderate, although the results in this thesis do not provide any conclusions about the scalability of the proposed solution. How many IPTV clients can be supported by one Retransmission Cache is yet to be investigated.

6.3 Discussion

The proposed design and implementation showed that packet retransmission can be applied successfully for a time-constrained IPTV service, distributed using a multicast IPTV network. The prototype evaluation showed that the memory requirements for the Retransmission Cache are moderate, and should not be a concern to apply in an access node. An IPTV client requires some some additional processing for the retransmission functionality, but the additional amount of memory needed should not cause a problem, due to the memory constraints that are lower bounded by the requirements of decoding and displaying of the video stream. This lower bound is much higher than the amount of buffer size that is required for enabling retransmissions. The upper bound for this buffer size is determined by the End-to-End delay requirements of the service, resulting in a trade-off between the application startup delay and the time available for the successful recovery of missing packets.

There are however two restrictions to the research described in this thesis. The first restriction is that the evaluation of the prototype did not contain any scalability experiments. These scalability experiments are necessary to determine how many IPTV clients can be supported simultaneously and if the access node would in practice be a good location to place a Retransmission Cache. Furthermore can these experiments show if the expected bottlenecks of the solution are the processing power and the transmission capacity of the access node.

The second restriction concerns the evaluation of retransmission mechanism in a congested network. The experiment results showed one possible drawback of enabling packet retransmissions in a congested network: they may contribute to additional packet loss. When the retransmission functionality is used in a network with additional, congestion aware traffic, the retransmission mechanism still leads to improvements of the perceived losses. There are however no results that evaluate the performance when congestion unaware services compete for the same bandwidth resources. An example would be the transmission of two IPTV streams, when the access network does not provide enough resources to support both. One can argue whether in a congested network the application of packet retransmissions or a Forward Error Correction for services that will not adapt the transfer rate to the congestion is a good solution. Especially for a service that requires a high Quality of Service like broadcast IPTV and thus requires network policing features that can assure this required quality.

6.4 Future work

One aspect that was already addressed previously is that the scalability of the proposed solution needs to be evaluated. This can provide insight in the application of a Retransmission Cache in an access node. For evaluation the possibilities of enabling multicast retransmissions can be taken into account, to determine under which circumstances it is more efficient to use multicast instead of unicast retransmissions.

Another research direction that was already discussed in the prototype evaluation chapter and in the discussion: when congestion occurs in the network, packet retransmissions may lead to higher packet loss rates. This may be prevented by using a scheduling mechanism to give retransmission packets a lower priority then packets from the IPTV stream. But this does not solve the problem when the congestion occurs in the home network environment. An IPTV service generally requires a high QoS level, and the only way to make sure that this can be achieved is offering QoS policing mechanisms in the home network, for instance by having resource reservation mechanisms or admission control in the home gateway.

A third topic for future work is investigating the possibilities to apply retransmissions based on the application payload. An example would be to only provide retransmissions for reference video frames and thereby reducing the bandwidth requirements for retransmissions.

Appendix A

Prototype experiment measures

The following parameters were measured during the experiments (RTX stands for retransmission):

Measure	Symbol	Definition
Experiment time	T_{exp}	The total execution time (in seconds) of a experiment run
Session bandwidth	BW	The bandwidth used to transmit the IPTV stream
Packets received	$RTPRecvd$	The number of IPTV stream packets received
RTX bandwidth	BW_{rtx}	The bandwidth used for packet retransmissions
RTX requests	$RTXReq$	The total number of retransmission requests. Per missing packet multiple retransmission requests can occur
RTX received	$RTXRecvd$	The number of RTX packets received
Used RTX packets	$RTXRecvdOK$	The number of RTX packets received and used
Discarded RTX packets	$RTXRecvdNOK$	The number of RTX packets received and discarded
Feedback messages	$RTXFB$	The number of feedback messages sent to indicate packet loss. One feedback message can be used to indicate the packet loss of up to 17 consecutive packets)
Packet unavailable rate	$P_{unavail}$	Percentage of retransmission requests that cannot be fulfilled
Client buffer size	$Buff_{client}$	The size of the IPTV client buffer (in bytes) before playback is started.
Client buffer fill level	$Buff_{clientAVG}$	The average number of bytes in the IPTV Client buffer.
Retransmission Cache buffer size	$Buff_{cache}$	The size of the Retransmission Cache buffer (in bytes)
Retransmission Cache buffer fill rate	$Buff_{cacheAVG}$	The average fill level of the buffer
RTX Round Trip Time	RTX_{RTT}	The time difference between sending a Retransmission Request for a specific packet and the reception of the respective packet.
Average RTX round trip time	$AVG(RTT_{RTX})$	The average RTX round trip time.
Delayed frames ratio	$Client_{DF}$	The percentage of decoded video frames that could not be displayed at the right moment

Table A.1: Prototype application measures

Bibliography

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for Real-Time applications," Internet Engineering Task Force, RFC 3550, Jul. 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3550.txt

[2] ITU-T focus group on IPTV, "IPTV Services Requirements," Working Document, ITU-T, Tech. Rep. FG IPTV-C-026, January 2007.

[3] "YouTube," 2008. [Online]. Available: www.youtube.com

[4] R. Good, "IPTV vs. Internet Television: Key Differences," Jun 2005, accessed March 25, 2008. [Online]. Available: http://www.masternewmedia.org/2005/06/04/iptv_vs_internet_television_key.htm

[5] "Apple TV," visited April 14th, 2008. [Online]. Available: http://www.apple.com/appletv/

[6] S. O'Hear, "Microsoft announces Internet TV for Windows Media Center," Sep 2007. [Online]. Available: www.last100.com/2007/09/27/microsoft-announces-internet-tv-for-windows-media-center-new-lineup-of-media-extenders/

[7] Wikipedia, "High-definition television — Wikipedia, the free encyclopedia," accessed 25-March-2008. [Online]. Available: "en.wikipedia.org/wiki/High-definition_television"

[8] L.-H. Chang, M.-Y. Liao, and J.-J. Liaw, "Scheduling Strategy for Realistic Implementation of Video on Demand over IPTV System," in *Proceedings of the 2007 WSEAS International Conference on Computer Engineering and Applications*, 2007.

[9] Motorola, "Time-Shifted Television - Expanding the Window of Television," July 2007. [Online]. Available: www.motorola.com/mot/doc/6/6531_MotDoc.pdf

[10] J. Postel, "Transmission control protocol," Internet Engineering Task Force, RFC 793, Sep. 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc793.txt

[11] ——, "User datagram protocol," Internet Engineering Task Force, RFC 768, Aug. 1980. [Online]. Available: http://www.rfc-editor.org/rfc/rfc768.txt

[12] E. Kohler, M. Handley, and S. Floyd, “Datagram congestion control protocol (dccp),” Internet Engineering Task Force, RFC 4340, Mar. 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4030.txt

[13] SDP Team, “MMS Streaming Protocol,” 2003, accessed Jan 01, 2008. [Online]. Available: http://get.to/sdp

[14] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, “RFC 4601: Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised),” RFC 4601 (Proposed Standard), Aug. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4601.txt

[15] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, “RFC 3376: Internet Group Management Protocol, Version 3,” RFC 3376 (Proposed Standard), Oct. 2002, updated by RFC 4604. [Online]. Available: http://www.ietf.org/rfc/rfc3376.txt

[16] B. Cohen, “Incentives build robustness in bittorrent,” 2003. [Online]. Available: www.bittorrent.org/bittorrentecon.pdf

[17] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, “Insights into pplive: A measurement study of a large-scale p2p iptv system,” in *Proc. of IPTV Workshop, International World Wide Web Conference*, 2006.

[18] “PPLive,” 2008. [Online]. Available: http://pplive.com

[19] “TvAnts,” 2008. [Online]. Available: http://www.tvants.com

[20] “SopCast,” 2008. [Online]. Available: http://www.sopcast.org

[21] Y. Huang, Y.-F. Chen, R. Jana, H. Jiang, M. Rabinovich, A. Reibman, B. Wei, and Z. Xiao, “Capacity analysis of MediaGrid: a P2P IPTV platform for fiber to the node (FTTN) networks,” in *Selected Areas in Communications, IEEE Journal on*, vol. 25, Jan. 2007, pp. 131–139.

[22] Y. Tang, L. Sun, M. Zhang, S. Yang, and Y. Zhong, “Live Video Streaming Service over Peer to Peer Network: Design, Implementation and Experience,” *International Journal of Computer Science and Network Security*, vol. 6, no. 3B, pp. 77–86, March 2006.

[23] H. Schulzrinne and S. Casner, “RTP profile for audio and video conferences with minimal control,” Internet Engineering Task Force, RFC 3551, Jul. 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3551.txt

[24] Internet Assigned Number Authority, “RTP Parameters,” Feb. 2008, last updated Feb 14th, 2008. [Online]. Available: http://www.iana.org/assignments/rtp-parameters

[25] B. d. Vleeschauwer, W. v. d. Meerssche, P. Simoens, and F. D. Turck, “On the enhancement of QoE for IPTV services through knowledge plane deployment,” *IST BroadBand Europe conference*, 2006.

[26] M. Handley, V. Jacobson, and C. Perkins, “SDP: Session Description Protocol,” Internet Engineering Task Force, RFC 4566, Jul. 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4566.txt

[27] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, “Hypertext transfer protocol – HTTP/1.1,” Internet Engineering Task Force, RFC 2616, Jun. 1999. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2616.txt

[28] M. Handley, C. Perkins, and E. Whelan, “Session announcement protocol,” Internet Engineering Task Force, RFC 2974, Oct. 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2974.txt

[29] H. Schulzrinne, A. Rao, and R. Lanphier, “Real time streaming protocol (RTSP),” Internet Engineering Task Force, RFC 2326, Apr. 1998. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2326.txt

[30] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “SIP: session initiation protocol,” Internet Engineering Task Force, RFC 3261, Jun. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3261.txt

[31] J. Rosenberg and H. Schulzrinne, “An RTP Payload Format for Generic Forward Error Correction,” RFC 2733 (Proposed Standard), Tech. Rep. 2733, Dec. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2733.txt

[32] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, “The secure real-time transport protocol (SRTP),” Internet Engineering Task Force, RFC 3711, Mar. 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3711.txt

[33] J. Ott, J.Chesterfield, and E.Schooler, “RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback,” Internet Engineering Task Force, Internet Draft, Expires: July 2008, Jan. 2008. [Online]. Available: http://www.rfc-editor.org/internet-drafts/draft-ietf-avt-rtcpssm-17.txt

[34] D. Komosny and V. Novotny, “Tree Structure for Specific-Source Multicast with Feedback Aggregation,” in *Networking, 2007. ICN ’07. Sixth International Conference on*, 22-28 April 2007, pp. 85–85.

[35] V. Novotny and D. Komosny, “Optimization of Large-Scale RTCP Feedback Reporting in Fixed and Mobile Networks,” in *Wireless and Mobile Communications, 2007. ICWMC ’07. Third International Conference on*, March 2007, pp. 85–85.

[36] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey, “RFC 4585: Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF),” RFC 4585 (Proposed Standard), Jul. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4585.txt

[37] J. Rey, D. Leon, A. Miyazaki, V. Varsa, and R. Hakenberg, “RFC 4588: RTP Retransmission Payload Format,” RFC 4588 (Proposed Standard), Jul. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4588.txt

[38] "Moving Picture Experts Group Home Page," MPEG Organization, accessed April 28, 2008. [Online]. Available: http://www.chiariglione.org/mpeg/

[39] "ITU-T Study Group 16," International Telecomunication Union, accessed April 28, 2008. [Online]. Available: http://www.itu.int/ITU-T/studygroups/com16/sg16-q6.html

[40] "H.263, H.264 MPEG4 information site," Video Technology Magazine, accessed April 278, 2008. [Online]. Available: http://www.h263l.com

[41] MPEG Industry Forum, "Understanding MPEG-4: Technologies, Advantages and Markets," MPEG Industry Forum, 2008. [Online]. Available: http://www.m4if.org/public/documents/vault/MPEG4WhitePaperV2a.zip

[42] "Divx," 2008. [Online]. Available: http://www.divx.com

[43] "Xvid," 2008. [Online]. Available: http://www.xvid.org

[44] "Apple quicktime," 2008. [Online]. Available: http://www.apple.com/quicktime/technologies/mpeg4/

[45] W. Soergel, "Scalable Video Coding." [Online]. Available: http://www.lnt.de/wsoergel/publications/scal_ARQ/node3.html

[46] M. Luby, A. Shokrollahi, and M. Watson, ""raptor forward error correction scheme for object delivery"," Internet Engineering Task Force, RFC 5053, oct 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5053.txt

[47] M. Johanson, "Adaptive Forward Error Correction for Real-time Internet Video," in *Proceedings of the 13th Packet Video Workshop, Nantes, France, April.*, 2003.

[48] R. Malik, A. Munjal, and S. Chakraverty, "Adaptive Forward Error Correction (AFEC) based Streaming using RTSP and RTP," in *Telecommunications, 2006. AICT-ICIW '06. International Conference on Internet and Web Applications and Services/Advanced International Conference on*, 19-25 Feb. 2006, pp. 60–60.

[49] S. Wenger, Y.-K. Wang, and T. Schierl, "RTP Payload Format for SVC Video, IETF Internet-Draft, Expires: August 24, 2008," Internet Engineering Task Force, Standards Track, 25 2008. [Online]. Available: http://tools.ietf.org/html/draft-ietf-avt-rtp-svc-08

[50] DSL Forum, "Triple-play Services Quality of Experience (QoE) Requirements," DSL Forum, Technical Report TR-126, 2006.

[51] A. Takashi, "Quality of Experience Requirements for IPTV," ITU-T, Working document FG IPTV-DOC-0086, 2007.

[52] "IP Performance Metrics Work Group," [Online; accessed 01-Januari-2008]. [Online]. Available: http://www.ietf.org/html.charters/ippm-charter.html

[53] K. Kerpez, D. Waring, G. Lapiotis, J. Lyles, and R. Vaidyanathan, “IPTV service assurance,” vol. 44, no. 9, pp. 166–172, 2006.

[54] “ITU-T Recommendation J.241: Quality of Service ranking and measurement methods for digital video services delivered over broadband IP Networks,” Tech. Rep., 2005.

[55] Telchemy Incorporated, “IPTV QoS/QoE Metrics,” ITU-T, Working document FG IPTV-C-0411, 2007.

[56] T. Friedman, R. Caceres, A. Clark, and Eds., “RTP control protocol extended reports (RTCP XR),” Internet Engineering Task Force, RFC 3611, Nov. 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3611.txt

[57] A. Clark and A. Pendleton, “RTCP XR - Video Metrics Report Blocks,” Internet Engineering Task Force, Internet-Draft draft-ietf-avt-rtcpxr-video-02, Nov. 2007, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-avt-rtcpxr-video-02.txt

[58] R. Babu, A. Bopardikar, A. Perkis, and O. Inge Hillestad, “No-reference metrics for video streaming applications,” in *Proc. of the 14th International Packet Video Workshop (PV2004)*. University of California Irvine, USA, Dec 2004.

[59] Telchemy Incorporated, “Understanding IP Video Quality Metrics,” Telchemy Incorporated, feb 2008. [Online]. Available: http://www.telchemy.com/appnotes/Understanding%20IP%20Video%20Quality%20Metrics.pdf

[60] “ITU-R Recommendation BT 500-7: Methodology for the subjective assessment of the quality of television pictures,” Tech. Rep., 1990.

[61] J. Klaue, B. Rathke, and A. Wolisz, “Evalvid - a framework for video transmission and quality evaluation,” in *Proc. of 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, Urbana, Illinois, USA, Sep. 2003, pp. 255–272.

[62] Telchemy Incorporated, “Proposed QoE algorithm based on PSNR Estimation,” ITU-T, Working document FG IPTV-C-0411, jan 2007. [Online]. Available: www.itu.int/md/T05-FG.IPTV-C-0413/en

[63] H. Lundqvist, T. Dietz, G. Nunzi, and M. Brunner, “IPTV over GPON - Function Study,” NEC Laboratories Europe, NEC Special Report NLE-SR-2007-53, June 2007.

[64] M. Prins, “IPTV Literature Study,” NEC Laboratories Europe, Tech. Rep. NLE-PR-2007-91, October 2007.

[65] S. Liew, S. Liew, and P. Wu, “A streaming-protocol retransmission scheme without client-server clock synchronization,” vol. 3, no. 7, pp. 223–225, 1999. [Online]. Available: http://www.ie.cuhk.edu.hk/fileadmin/staff_upload/soung/Journal/J8.pdfs

[66] P. Karn and C. Partridge, “Improving round-trip time estimates in reliable transport protocols,” in *SIGCOMM ’87: Proceedings of the ACM workshop on Frontiers in computer communications technology*. New York, NY, USA: ACM, 1988, pp. 2–7.

[67] Videolan, “VLC Media Player",” 2007. [Online]. Available: http://www.videolan.org/

[68] W. Ali and A. Nigam, “Platform independent RTP implementation in Java,” 1998. [Online]. Available: http://www.cs.columbia.edu/~hgs/teaching/ais/1998/projects/java_rtp/report.html

[69] H. Schulzrinne and J. Lennox, “RTPlib - RTP library,” 2000. [Online]. Available: http://www-out.bell-labs.com/project/RTPlib/

[70] “FMPEG,” 2008. [Online]. Available: http://ffmpeg.mplayerhq.hu/

[71] “SDL - Simple Directmedia Layer.” [Online]. Available: http://www.libsdl.org"

[72] “NIST/SEMATECH e-Handbook of Statistical Method,” National Institute of Standards and Technology, July 2006, [accessed June 13, 2008]. [Online]. Available: http://www.itl.nist.gov/div898/handbook/eda/section3/eda352.htm

[73] S. Hemminger, “Network emulation with netem,” in *Proceedings of the Australia’s National Linux Conference (LCA ’05)*, 4 2005.

[74] A. Keller, “TCN Trace Control for Netem,” 2008. [Online]. Available: http://tcn.hypert.net/

[75] S. Avallone, S. Guadagno, D. Emma, A. Pescapè, and G. Ventre, “D-ITG Distributed Internet Traffic Generator,” in *QEST*. IEEE Computer Society, 2004, pp. 316–317. [Online]. Available: http://dblp.uni-trier.de/db/conf/qest/qest2004.html#AvalloneGEPV04

[76] “Octave.” [Online]. Available: http://www.gnu.org/software/octave/

[77] “Wireshark.” [Online]. Available: http://www.wireshark.org

List of Figures

List of Tables

Glossary

Notation	Description
ACK	Acknowledgement
codec	compression/decompression
CSRC	Contribution Source
DCCP	Datagram Congestion Control Protocol
DSLAM	Digital Subscriber Line Access Multiplexer
DVB	Digital Video Broadcasting
EPSNR	estimated PSNR
FB	Feedback
FCI	Feedback Control Information
FEC	Forward Error Correction
GOP	Group Of Pictures
HD	High Definition
HG	Home Gateway
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IGMP	Internet Group Membership Protocol
IPPM	IP Performance Metrics
IPTV	Internet Protocol Television
ITU	International Telecommunication Union
ITU-T	ITU Standardization Sector

Notation	Description
MMS	Microsoft Media Server
MOS	Mean Opinion Score
MPEG	Moving Picture Experts Group
MPEG-TS	MPEG Transport Stream
MSAN	Multi Service Access Node
NACK	Negative Acknowledgement
NVOD	Near Video on Demand
P2P	Peer to Peer
PIM-SM	Protocol Independent Multicast - Sparse Mode
PLR	Packet Loss Rate
PSNR	Peak Signal to Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
RTT	Round Trip Time
RTX	Retransmission
SAP	Session Announcement Protocol
SD	Standard Definition
SDL	Simple DirectMedia Layer
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SSRC	Synchronization Source
STB	Set Top Box
SVC	Scalable Video Coding
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VCEG	Video Coding Experts Group
VOD	Video on Demand
VoIP	Voice over IP

Notation	Description
VQM	Video Quality Metrics
XR	Extended Report