TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Miikka Lundan

Packet-Switched Streaming Service in Non-Bitrate-Guaranteed Mobile Networks

Julkaisu 969 • Publication 969

Tampere 2011

Tampereen teknillinen yliopisto. Julkaisu 969
Tampere University of Technology. Publication 969

Miikka Lundan

Packet-Switched Streaming Service in Non-Bitrate-Guaranteed Mobile Networks

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB104, at Tampere University of Technology, on the 3rd of June 2011, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2011

ISBN 978-952-15-2580-3 (printed)
ISBN 978-952-15-2594-0 (PDF)
ISSN 1459-2045

Abstract

Streaming allows consuming content, such as video or audio, without storing it on a receiving device. The benefit of streaming is that it requires little memory from the device. The drawback is that the device must always be connected to the network and that the network connection must offer adequate capacity and performance stability. These network connection requirements pose a number of challenges especially for mobile devices connected to packet-switched networks, in which the basic network connections cannot guarantee the allocated bandwidth or limit delays under certain levels.

A comprehensive study and a large set of simulation tests were carried out to analyze the characteristics of packet-switched networks that cannot guarantee the capacity. In this dissertation, such networks are referred to as "*non-bitrate-guaranteed mobile networks*". This study revealed several key problems including: data transfer gaps caused by cell reselection, bandwidth fluctuation caused by either signal quality or cell load, streaming client buffer underflow and limited capability of the streaming client to report these problems to the streaming server. Novel solutions to these problems are proposed in this thesis. In addition to these service issues, this thesis evaluated subjective quality thresholds of lip synchronization and video. It is equally important, compared to technical improvements, to understand whether the improvement give subjective benefits. For example, increased bitrate improves the video/audio quality of streaming service, but the device hardware may become limiting factor that improvements are no longer noticed.

Four fundamental problems were investigated in the thesis. The first problem is related to cell reselection. This is a natural phenomenon of every mobile network, but unfortunately, it has a tendency to cause gaps in the dataflow, which causes noticeable quality degradations in media streaming. This thesis provides a novel solution to hide the cell reselection from users and proposes a novel mechanism to maintain a full streaming client buffer. The proposed solution makes use of standard Packet-switched Streaming Service (PSS) methods and requires no changes to the network.

The second fundamental problem this thesis examines is related to how network capacity may change if the mobile device moves into a position that changes the received signal quality. Another major factor affecting network capacity is the amount of devices served by the network cell. The change in capacity can be positive or negative. Streaming service users are unable to detect positive capacity change from the streaming client, which simply means that the network could offer better service than what is currently being provided. Negative capacity change, on the other hand, is easily detectable by the user. When the streaming client buffer underflows, the quality drops drastically. The thesis provides a novel solution for the streaming client to report network capacity changes to the streaming server. Again, an additional merit of the proposed solution is that it uses standard Packet-switched Streaming Service (PSS) methods and requires no changes to the network.

The range of options for reporting streaming service problems or simple statistical characteristics is very limited. Real-time Transfer Control Protocol (RTCP) offers a limited set of predefined parameters that allow reporting problems. This is the third fundamental problem investigated in the thesis and a novel solution for reporting problems and statistical information from client to server is proposed. The method allows the client to provide raw data to the streaming server, which may include relevant statistics about the quality of the service. It also offers a parameter negotiation technique allowing the client to easily expand the current set of parameters used and a better adaptation to different types of client devices.

When improving streaming services in mobile devices, it is important to ensure that such improvements are noticeable by the user. Due to the limited size of a mobile device display, improvements in content streaming may not be visible and prior measurements in e.g. the television industry may not apply. This fourth problem is investigated in this thesis and new parameters specific to mobile devices are determined. The first objective is to determine the bounds within which synchronization of audio and video (i.e. lip synchronization) may vary and still produce acceptable results for the user. The second objective is to determine the correct ranges of bitrate and

frame rates that should be used by service providers in order to guarantee a certain streaming content quality. A comprehensive study was carried out and suitable parameters and ranges are proposed for mobile streaming applications.

The main emphasis in this study is on the *Non-bitrate-guaranteed networks,* which illustrate the basic and most limited mobile networks existing at the time the research was carried out. Most of these issues mentioned above have already been solved in more advanced networks, but such networks have limited abilities to solve such problems for all users, since it may be too costly to offer them to all users. The elegant solutions described herein require no changes to the network; therefore, availability would not be limited by the network. If the service provider chooses to implement these in mobile client and service servers, they would be available for all users regardless of the underlying network.

Preface

The research presented in this thesis has been carried out in Nokia Corporation, Tampere, Finland.

First and the foremost I wish to express my profound gratitude to my supervisor, Professor Moncef Gabbouj of Tampere University of Technology for his continuing support, friendly guidance and valuable comments. His encouragement and assessment provided me confidence and realistic hope.

I would like to thank Professor Markus Rupp from Vienna University of Technology and Assistant Professor Jari Korhonen from Technical University of Denmark, the reviewers of this thesis, for their critical but constructive opinions and competent judgment.

In addition I would like to thank Igor Curcio, from Nokia Corporation, for endless hours while performing the tests and publishing the results, and giving confidence that I should also start writing doctoral thesis.

Over the years I have had the privilege to work with a wonderful group of people, who have helped with this study. I thank all of them.

Special thanks to my wife Dr. Arja Lundan for endless love and support. And thanks to my son Andrey whose smile is an everlasting source of energy and delight.

Last but not least, I wish to express my warmest thanks to my parents Mrs. Ritva Lundan and Mr. Viljo Lundan, who have always encouraged me to study further. I can only wish that my father would have seen the final moments of my studies.

Tampere, June 2011 Miikka Lundan.

Contents

List of Pubications

The compound thesis, including a summary part and a set of publications, is written on the basis of the following publications. In the summary part, these publications are referred to as Publications [P1], …, [P8]

[P1] **Miikka Lundan** and Igor Curcio, "3GPP streaming over GPRS Rel. '97," *Proc. 12th IEEE International Conference on Computer, Communications and Networks (*ICCCN '03), 20-22 October 2003, Dallas, TX, U.S.A., pp. 101-106.

[P2] Igor Curcio and **Miikka Lundan,** "Optimal 3GPP Packet-switched Streaming Service (PSS) over GPRS networks," *Multimedia tools and application*, Volume 35, Number 3, December 2007.

[P3] **Miikka Lundan,** "Streaming over EGPRS*," Proc. IEEE 9th Symposium on Computers and Communications* (ISCC04), 28 June - 2 July 2004, Alexandria, Egypt, pp. 969-974.

[P4] **Miikka Lundan** and Igor Curcio, "Mobile streaming service in WCDMA network*,"* Proc. IEEE 10th Symposium on Computers and Communications (ISCC05), 27-30 June 2005, Cartaghena, Spain, pp. 231-236.

[P5] Igor Curcio and **Miikka Lundan**, *"*Event-driven RTCP Feedback for mobile multimedia applications*," Proc. 3rd Finnish IEEE Wireless Communications Workshop* (FWCW'02), 29 May 2002, Helsinki, Finland, pp 1-2.

[P6] Igor Curcio, Juha Kalliokulju and **Miikka Lundan**, *"*AMR mode selection enhancement in 3G networks,*" Multimedia Tools and Application*, Volume 28, Number 3, March 2006.

[P7] Emre Aksu, Igor Curcio, David Leon, **Miikka Lundan,** Viktor Varsa, and Ru-Shang Wang, "Bandwidth adaptation," *Patent application WO2004/028095 (April 2004), Granted patents FI116498 (November 2005), CN1314247 (May 2007), EG23825 (September 2007) and HK1084268 (October 2007).*

[P8] Igor Curcio and **Miikka Lundan**, "Human perception of lip synchronization in mobile devices," *Proc. IEEE Symposium on a World of Wireless, Mobile and Multimedia Networks* (WOWMOM 2007), 18-21 June 2007, Helsinki, Finland.

List of Acronyms

3GPP	Third Generation Partnership Project
AAC	Advanced Audio Codec
AAC-LC	AAC Low Complexity
AAC-LTP	AAC Long Term Prediction
AMR	Adaptive Multirate
AMR-WB	AMR-Wideband
ATSC	Advanced Television System Committee
AVC	Advanced Video Codec
BER	Bit Error Rate
BSC	Base Station Controller
BTS	Base Transceiver Station
CC/PP	Composite Capability/Preference Profile
C/I	Carrier/Interference
CS	Coding Scheme
DIMS	Dynamic and Interactive Multimedia Scenes
DLS	DownLoadable Sounds
DRM	Digital Rights Management
GERAN	GSM EDGE Radio Access Network
GIF	Graphics Interchange Format
GPRS	General Packet Radio Service
GTP	GPRS Tunneling Protocol
eAAC	Enhanced AAC
ECSD	Enhanced Circuit Switched Data
EDGE	Enhanced Data rates for Global Evolution
EGPRS	Enhanced GPRS
FLO	Flexible Layer One
HTTP	Hyper Text Transfer Protocol
IMS	Internet Multimedia Subsystem

IP	Internet Protocol
ISHO	Inter-System Handover
ISO	International Organization for Standardization
JFIF	JPEG File Interchange Format
JPEG	Joint Photographic Experts Group
LLC	Logical Link Control
MBMS	Multimedia Broadcast Multicast Service
MCS	Modulation and Coding Scheme
MDLS	Mobile Downloadable Sounds
MIDI	Musical Instrument Digital Interface
MIME	Multipurpose Internet Mail Extensions
MMA	MIDI Manufacturers Association
MPEG	Moving Picture Experts Group
MXMF	Mobile eXtensible Music Format
NACC	Network Assisted Cell Change
NALU	Network Abstraction Layer Unit
OMA	Open Mobile Alliance
PDP	Packet Data Protocol
PNG	Portable Networks Graphics
PSNR	Peak-Signal-to-Noise Ratio
PSS	Packet-switched Streaming Service
QoE	Quality of Experience
QoS	Quality of Service
RDF	Resource Description Framework
RLC	Radio Link Control
RNC	Radio Network Controller
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
SAIC	Single Antenna Interference Cancellation
SDP	Session Description Protocol
SDU	Service Data Unit
SGSN	Serving GPRS Support Node

SMIL	Synchronized Multimedia Integration Language
SNR	Signal-to-Noise Ratio
SP-MIDI	Scalable Polyphonic Musical Instrument Digital Interface
SQCIF	Sub Quarter Common Interface Format
SRTP	Secure RTP
SVG	Scalable Vector Graphics
TBF	Temporary Block Flow
TFRC	TCP Friendly Rate Control
TCP	Transmission Control Protocol
TS	Timeslot
UAProf	User Agent Profile
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System
URI	Uniform Resource Identifier
UTRAN	UMTS Terrestrial Radio Access Network
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WCDMA	Wideband Code Division Multiple Access
WWW	World Wide Web
XHTML	eXtensible Hyper Text Markup Language
XMF	eXtensible Music Format

List of Tables

List of Figures

Note: All the pictures of mobile devices are published with the permission of the Nokia Corporation. Figure 7 and Figure 8 are published with the permission of Strathclyde University, Glasgow.

Chapter 1

Introduction

PACKET- switched connection is based on Internet Protocol (IP), and in the mobile environment, it was first introduced by General Packet Radio Service (GPRS). In packet switching, the data is divided into packets, each packet having an identifier or address, which is used by routers in the network to pass the packet to its intended destination. Streaming is one of the services using packet-switched network technology. It is an on-demand media consumption method, in which a user receives data through the network while playing it. Compared to traditional downloads the greatest difference from end-user point of view is that in streaming, the content consuming experience is impacted by the network changes and problems of the network are visible to the end user almost immediately. Another major difference is that the content is not saved or stored on the device, thus it doesn't require much memory, but cannot be consumed without network access. Streaming players allow basic playback methods such as Play, Pause, Rewind and Fast-Forward, and players typically buffer a couple seconds of media to handle data packet arrival fluctuation.

Packet-switched Streaming Service (PSS) is a Third Generation Partnership Project (3GPP) standard that describes the codecs and protocols to be used in a packet-switched-based streaming service. The first version (called Release 4) was released in 2000 and the latest release in 2010 is Release 9 (All Real-Time Streaming Protocol (RTSP) related changes have been introduced previously in Release 7. Releases 8 and 9, add more details to Hyper Text Transfer Protocol (HTTP) streaming, which was omitted in this thesis). PSS standard allows the service providers and client manufacturers to devise end-to-end services that work with many clients.

1.1. NON-BITRATE-GUARANTEED MOBILE NETWORK

One of the basic differences between packet switched and circuit switched networks is that in packet switched networks, a bandwidth allocated to the data transmission may not be guaranteed, in addition the packet arrival time (the time between two consecutive packets) may vary greatly. In this thesis, the *non-bitrate-guaranteed* term is used to describe the network with these two characteristics. The majority of packet-switched network methods fall into this category, since they do not guarantee the network bandwidth that was given at the beginning of the session, and the packet delay has no bounds. Also if an end-user is mobile, the cell reselections cause data gaps in streaming. GPRS falls in this category as well as Enhanced GPRS (EGPRS) until GERAN5 (after that it starts to follow the WCDMA traffic classes). The Wideband Code Division Multiple Access (WCDMA) network is divided into four data traffic classes, where Background and Interactive classes are non-bitrate-guaranteed and Streaming and Conversational classes are bitrate guaranteed.

Although different technologies have been readily available for a decade, commercial streaming services for mobile devices remain rare. There are three reasons that may explain why such streaming has not been a major success. First even minimally adequate media quality requires a network that provides to the user a steady tens-of-kilobits-per-second bandwidth. Until recently this has not been widely available. Second, device capabilities have only recently reached an acceptable level in mass-market devices. Nowadays, most mobile devices support adequate networks and codecs. Third reason is purely cultural: people simply do not see the need to watch video from their mobile device. Mobile TV may change this cultural view and make other video services such as streaming more popular. When people learn one mode of watching video, they do not know the underlying technologies, and may easily switch to use other services that use different technologies.

1.2. SCOPE OF THE THESIS

This thesis began with a simple question: *Is streaming service possible in a GPRS network?* Once it was determined that the answer to this question was *yes*, the next question that arose was *what existing GPRS parameters should be used to provide the best possible quality?* It did not take long to realize that the proposed parameters are not the best for the service and that there are some core problems that cannot be solved with the current standards. The next question was *what improvements are needed to make the session better?* The core problems that need to be solved were cell reselection and bitrate fluctuation and this thesis offers novel solutions to both of these.

When the EGPRS networks were released, a study was done aiming to determine *what improvements does it brings to the service?* Increased bitrate and improved error handling naturally improved the visual quality, and the experience as a whole, but the core problems found in GPRS (cell reselection and bitrate fluctuation) still remained.

The final testing round was done after WCDMA networks were made available. In WCDMA, the cell reselection and bitrate fluctuation problems were partly solved, since some cell reselections were lossless (but not all) and a fixed bitrate is guaranteed (if the right service class was available). It should be noted that the core problems (cell reselection and bitrate fluctuation) still exist in WCDMA, but they occur much more rarely than in the GPRS or EGPRS environment. After the basic understanding of the nature of the PSS in mobile networks was gained, a phase was initiated to identify the ways to improve the service.

1.3. OUTLINE OF THE THESIS

This thesis is structured as follows. Chapter 2 describes video streaming and provides an overview of the Packet-switched Streaming Service (PSS) and how it has evolved throughout the different releases of the standard. Chapter 3 takes a closer look at packet-switched networks (GPRS, EGPRS and WCDMA) and their Quality of Service parameters are given. The focus is not to thoroughly describe the network technologies, but to concentrate on those features that involve streaming services. In Chapter 4, the basic characteristics of PSS are shown using the test results. This chapter also describes the testing methods used and provides a literature review of previous studies. Chapter 5 describes the proposed improvements and compares these to other methods. Chapter 6 briefly describes some subjective quality aspects of mobile media consumption. Key focus areas are lip synchronization and video quality. Chapter 7 concludes the thesis.

1.4. AUTHOR'S CONTRIBUTIONS TO THE PUBLICATIONS

Author's contribution to the publications is described in the following.

In Publications [P1 and P4], Author was the main author. Author ran the test and analyzed the results revealing the recommended Quality of Service (QoS) parameters and issues with cell reselection and bandwidth adaptation. Papers were written in very close collaboration with the co-author (Igor Curcio) and Author performed about 80% of the total amount of work required (including testing, analyzing and publishing) on P1 and 60% on P4.

In Publications [P2 and P8] Author was a co-author, performing all the tests and analyzing the results. [P2] increased the basic knowledge of streaming over GPRS and proved the efficiency of proposed cell reselection management method. [P8] revealed new lip synchronization thresholds for mobile devices. Papers were written in very close collaboration with the main author (Igor Curcio) and Author contributed to all chapters of the papers. Author performed about 60% of the total amount of work required.

In Publication [P3] Author was the only author, performing the tests, analyzing results, and writing the article alone. Publication increased the basic knowledge of streaming over EGPRS.

In Publications [P5] Author was a co-author running the tests and analyzing the results. Publication allowed understanding the limitations of event-based RTCP feedback mechanism. Paper was written in very close collaboration and Author contributed to the chapters that described the testing, experimental results and conclusions. Author performed about 80% of the total amount of work required.

In Publication [P6] Author was the co-author. The original idea of network based Adaptive Multirate (AMR) mode selection is from the two main authors of the article (Igor Curcio and Juha Kalliokulju). The Author tested this idea and proposed some changes to the original idea (the biggest being the weighting factor in the packet loss computation calculation, which allows higher value weights in the latest values, compared to older values). The Author's role was to run and analyze the tests and write the Simulation Results chapter. Author performed about 30% of the total amount of work required.

In Publications [P7] there was no main author or co-authors. Innovation of bandwidth adaptation was created in brainstorming sessions in which Author participated. It is impossible to say who was the main inventor, but the Author's role in these sessions was significant.

Chapter 2

Packet-switched Streaming Service

THE first Packet-switched Streaming Service specification [3GPP-4 26.233] defined streaming as the *"ability of an application to play synchronized media streams like audio and video streams in a continuous way while those streams are being transmitted to the client over a data network."*

To assist in understanding this complex sentence, Table 1 compares and contrasts streaming to more common playback and file transfer methods.

Table 1: Data transfer and playback methods

Method	**Saved file**	**Playback during transfer**	**Playback controls**	**Low memory consumption**	**On demand service**
Download	Yes	No	Yes	No	Yes
Progressive download	Yes	Yes	Yes	No	Yes
Broadcasting (TV)	No	Yes	No	Yes	No
Streaming	No	Yes	Yes	Yes	Yes

Traditional downloading is the most common way to transfer a file over the IP network. There a user selects a file, waits until it is fully transferred to the device and afterwards, a user can start the playback. The file is stored on a device and therefore it can consume a great deal of memory, but the user can access it locally. Progressive downloading adds the possibility of beginning the playback during file transfer, but after the downloading phase is completed it acts the same way as a traditional download. Broadcasting is much closer to streaming than downloading. In broadcasting, a file is not saved and playback occurs during data transfer. What streaming adds to broadcasting is that a user can control the playback (wind the stream, pause and stop the playback), and it also allows the user to start the playback on demand without broadcasting schedules. The drawback of streaming compared to downloading is that consuming the content always requires a network connection, which may not be totally reliable. The benefit is that the device does not have to have an enormous amount of memory to keep the data. A small buffer (a few seconds), is enough to handle packet arrival jitter. One can think of video streaming as similar to a video rental. The user can choose the time they watch the video, and during the playback, the user has full control of the video, but the video is not theirs to keep, so the user must “return” it to the service (there are streaming clients that acts like progressive downloading, but the ideology of streaming is not to store the data). Specifications also enable live streaming where the user joins a more broadcasting-like type of transmission, and is for example, able to pause the transmission and start the playback later.

2.1. OVERVIEW OF PSS

A packet-switched streaming service (PSS) is a set of specifications defined by the Third Generation Partnership Project (3GPP). The first version of PSS was called Release 4, and the latest version is Release 9 (Since Release 8 and 9 concentrate on HTTP streaming, these are omitted from this thesis). A quick overview of each release can be found in Figure 1. The upper right triangle shows the main features of the releases, while the lower left triangle lists the supported media formats. Note that not all listed features and media formats are mandatory. Only the underlined items in Figure 1 are mandatory in each release.

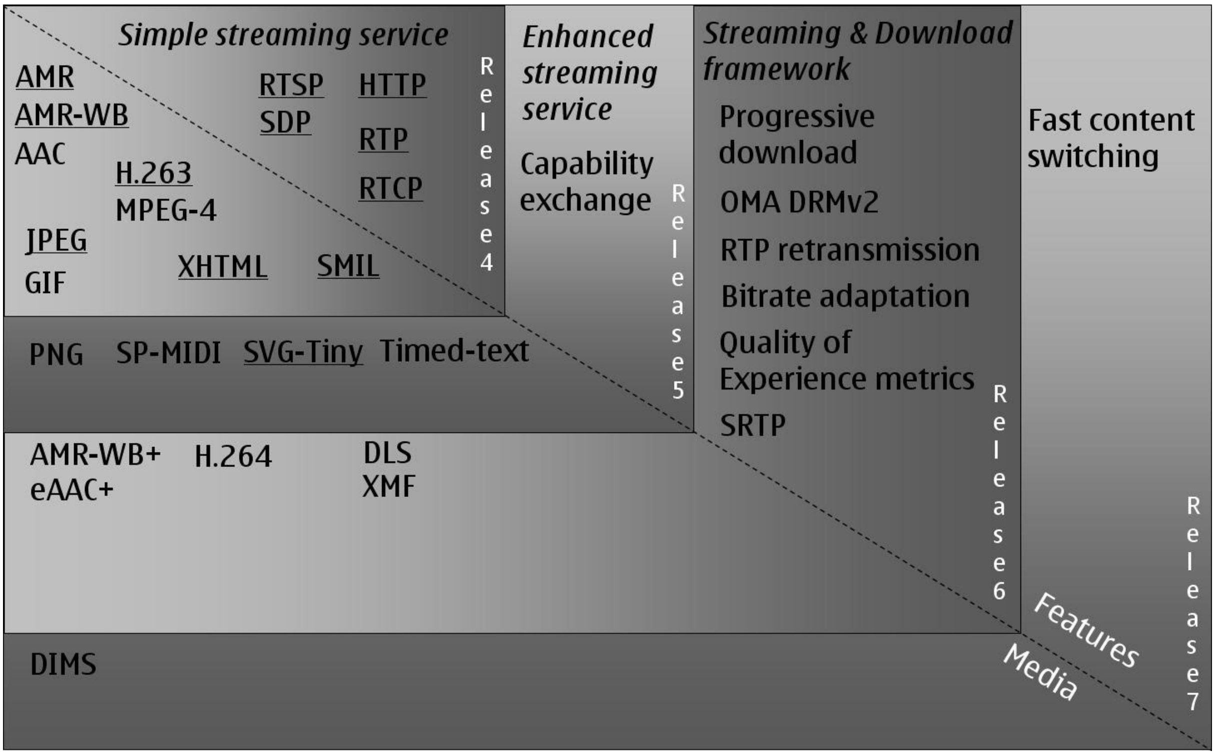


Figure 1: PSS overview

The minimum set of features, protocols and codecs required to establish a streaming service (called *Simple streaming service*) is defined in PSS Release 4 [3GPP-4 26.233 & 3GPP-4 26.234]. *Enhanced streaming service* was introduced in PPS Release 5 [3GPP-5 26.233 &

3GPP-5 26.234], which in practice means adding capability exchange to the *Simple streaming service*. PSS Release 5 introduced one new specification, Stage 1, which describes the overall service requirement (backward compatibility, dynamic updates, Quality of Service, service personalization, security and charging issues) [3GPP-5 22.233]. *Streaming and the download framework* is described in PSS Release 6 [3GPP-6 26.233]. The framework includes progressive downloading, content encryption, bitrate adaptation, Quality of Experience metrics, Real-time Transport Protocol (RTP) retransmission and a strengthened capability exchange mechanism. PSS Release 7 [3GPP-7 26.234] is a minor update including mainly clarifications and corrections. Release 7 includes two new features: Fast content switching and Dynamic and Interactive Multimedia Scenes (DIMS) [3GPP TS 26.142]. DIMS offers an additional scene description method to SMIL.

PSS is a client-server-based service, where the server may reside inside the operator's network or in the public Internet. The server and client communicate with each other through the mobile IP network. The streaming client must be designed for mobile use. Data transport can be either over an RTP using a User Datagram Protocol (UDP) for continuous media (video, audio, speech and timed text) or Hyper Text Transfer Protocol (HTTP) using Transmission Control Protocol (TCP) for discrete media (scene descriptions, still images, graphics, text, timed text and synthetic audio). Session set-up and control can occur using both protocols depending on media types. Real-Time Streaming Protocol (RTSP) is used for RTP traffic, while HTTP is used for discrete media traffic. Figure 2 illustrates the protocol stack of PSS [3GPP-6 26.233].

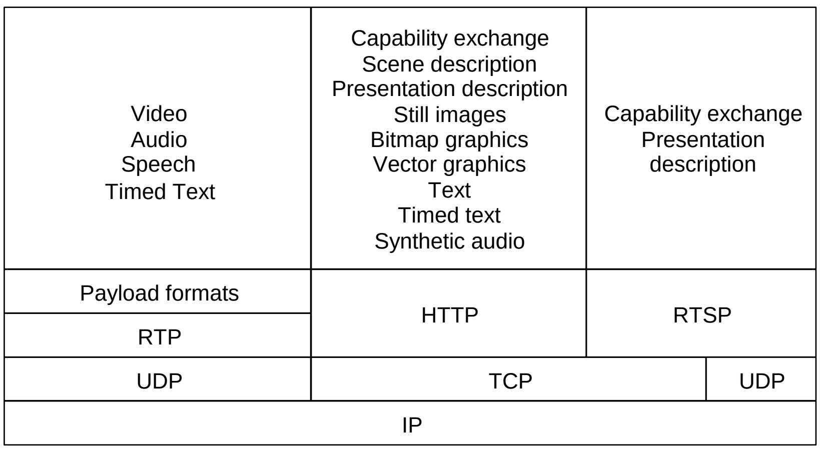


Figure 2: PSS Protocol stack

PSS client consists of different types of codecs and session management elements. Figure 2 in [P2] shows the functional components of the PSS client described in [3GPP-6 26.233]. It also describes what the scope of the PSS specification is and what is left out. (E.g. media synchronization and playback are left out of the PSS specification.)

Real-time Streaming Protocol (RTSP) and Session Description Protocol (SDP) are the key session setup protocols that will be described in Section 2.2. Real-time Transfer Protocol (RTP) and Real-time Transfer Control Protocol (RTCP) handle the data transfer described in Section 2.3. The entire media "triangle" of Figure 1 is described in the Section 2.4.

2.2. STREAMING SESSION

Before the streaming session can start, the device must establish a network connection. During the connection setup, the device can request certain Quality of Service (QoS) parameters that characterize the service. The details of these QoS parameters are described in Section 3.2, and Section 5.1 describes the findings of appropriate QoS parameters for streaming services.

Streaming session itself can be divided into three parts:

1. Session setup
2. Possible session changes during ongoing session
3. Session teardown

All of these three session parts require a Real-Time Streaming Protocol (RTSP) [RFC 2326] to manage the session. In the session setup, the Session Description Protocol (SDP) [RFC 4566] is used for describing the media components and their parameters. The session is established using a Transmission Control Protocol (TCP) [STD 0006], which allows the retransmission of lost session establishment packets. The minimal RTSP implementation is defined in the PSS Release 4 [3GPP-4 26.234]. Typical Streaming session RTSP message flow is shown in Figure 4 of [P2].

2.2.1. Session setup

A *Simple streaming service* defined by PSS Release 4 allows the user to start the session in three different ways:

1. In a World Wide Web (WWW) or Wireless Application Protocol (WAP) browser
2. The user writes streaming URI by hand
3. The user obtains an SDP file (e.g. through MMS)

A web-browser is the most typical software application used to begin the streaming session. The session setup starts with an RTSP Describe message, which basically includes only the URI of the requested content. The server responds to the requests by sending an SDP body that describes the content. With a basic SDP, it is possible to define the codecs, bitrates used and length of the sequence. The SDP also allows quite free extensions that can be used to negotiate many parameters. In Chapter 5, some novel extensions are described. It is possible that a session contains only non-streamable content like an SMIL file or time-synced presentation and such sessions do not require an SDP file. Since this thesis concentrates on video streaming, such sessions are not described. PSS Release 4 requires that all mandatory fields of SDP are used and also that Appendix C of the RTSP specification is followed, which defines the usage of SDP fields in an RTSP session. In addition to these, it requires that certain SDP attributes (such as bandwidth and range) are used in order to control and describe the media better.

If a client is capable of receiving the content, it requests to setup the media. At this phase the server gives the session a unique identification and negotiates the transmission ports used. An RTSP Setup request must be done once per media type. After a successful media setup, the client requests to begin the media transfer with RTSP Play, and the dataflow begins.

New buffering related SDP attributes were introduced in Release 5. Annex G of [3GPP-5 26.234] describes the buffering parameters that allow the streaming client to optimize the video buffer size according to the video bitrate of the stream. Typically, the streaming client has a constant video buffer size that is either the result of a compromise between different networks or oversized according to the worst-case scenario. If the bitrate varies too much, the assumptions at the beginning of the session are no longer valid, and the buffer can overflow or underflow. The target of these new parameters is to create a dynamic buffer size that enables a pauseless playback if the dataflow is within certain boundaries.

Capability exchange functionality, introduced in PSS Release 5 [3GPP-5 26.234], allows the server to request device capability information listed below:

- Audio (channel and polyphony), Video (bitrate and buffer size)
- MIME-types (Multipurpose Internet Mail Extensions)
- Vendor, Model, PSS and SMIL version
- Device Screen details
- Languages, characters and encoding

With these details, a server can tailor the content to suit the device capabilities, and this also allows a server to handle devices with different PSS versions. Capability exchange URLs are recommended to be included in the RTSP Describe request, but it can be sent in any RTSP request. Based on these URLs (There can be more than one), a server can fetch the data from the device profile server. A device can also provide additional information that clarifies or overrides some information received from the server. A device profile server sends a device capability profile message that is Resource Description Framework (RDF) [W3C RTF] document that follows the structure of the Composite Capability/Preference Profile (CC/PP) framework [W3C CC/PP] and CC/PP application User Agent Profile (UAProf) [WAG UAProf].

Capability exchange was further enhanced in Release 6 [3GPP-6 26.234]. Many new features introduced in Release 6 received their own parameter to define whether the feature is supported or not (see Section 2.3 for more information about these features.):

- Link characteristics
- Bandwidth adaptation
- Quality of Experience (QoE)
- RTP retransmission and extended RTCP reporting
- Digital Rights Management

Fast content switching introduced in Release 7 [3GPP-7 26.234] improves start-up and content switching by reducing the interaction between client and server. RTSP messages can be pipelined together to reduce the message-response cycles. For example, combining two Setup and Play messages into one message reduces the handling time. In content switching, the improvement is achieved by using the former media components (e.g. ports) as much as possible so that not everything needs to be negotiated from the beginning.

2.2.2. Changes during the session

During the session, a user can do some basic changes to a video stream such as choose Pause, Play (after pause), Rewind and Fast-Forward with RTSP messages. The client and server can also change information during the session and change some parameters without notifying the user. The bandwidth adaptation described in Release 6 can make changes to the session. The Quality of Experience report can be included in an RTSP message and may introduce a change to the session.

2.2.3. Session teardown

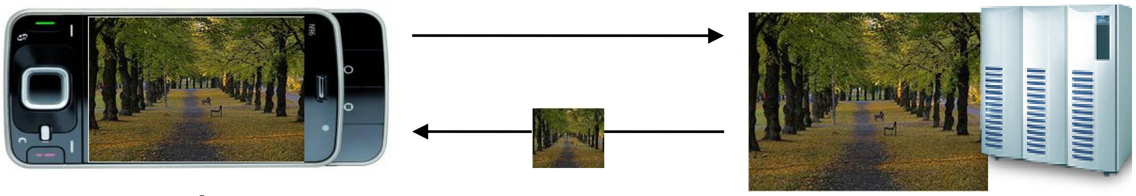
A streaming session is stopped with an RTSP Teardown message. Both client and server can initiate a teardown. A server-initiated teardown indicates the end of the stream, while a client-initiated teardown is a voluntary cancellation of the session. The last QoE report can be included in a Teardown message.

2.3. MEDIA TRANSMISSION

When the streaming session is established, the media transmission can begin. The PSS Release 4 [3GPP-4 26.234] specifies the basic media transmission methods and protocols including a simple feedback. PSS Release 6 [3GPP-6 26.234] was a major update to this area, since it introduced adaptive- and secure data transmission and a more advanced feedback mechanism.

2.3.1. Basic data transmission

Continuous data transmission (i.e. audio and video) is handled with a Real-time Transport Protocol (RTP) [RFC 3550] over a User Datagram Protocol (UDP) [STD 0006]. RTP allows packetizing the data with source identifiers, timestamp and a sequence number that enables a streaming client to build a continuous media stream out of several packets. Typically audio and video data is transferred in different packets making synchronizing the media back together an important task. Since the UDP does not guarantee lossless transmission, some media packets may be lost during transmission. Figure 3 illustrates the elements of basic data transmission. A streaming client first requests content from the server, and then the server streams the content to the client.



Streaming client

Streaming server and content

Figure 3: Basic streaming

The RTP specification also includes a feedback mechanism with a Real-time Transmission Control Protocol (RTCP). Basic RTCP allows the client (receiver) and server (sender) to report packet losses and packet arrival jitter. The sending interval of RTCP packet is limited. It is recommended to send an RTCP packet every five seconds, but packets cannot take more than 5% of the total bandwidth (2.5% for sender reports and 2.5% for receiver reports). With low bandwidth, it may not be possible to follow these recommendations. This issue is presented further in Section 5.2.1

Discrete media (i.e. images, graphics, text and scenes) are transmitted with Hypertext Transfer Protocol (HTTP) [RFC 2616] over a Transmission Control Protocol (TCP) [STD 0007].

PSS Release 5 defines streaming service requirements [3GPP-5 22.233], which state that a client shall be able to request appropriate QoS levels related to the service. The exact details as to how this should be done and what are appropriate QoS levels are left open. The findings on appropriate QoS levels can be found in Section 5.1

Progressive downloading, introduced as a part of the PSS in Release 6, is HTTP-based downloading over TCP. The difference with traditional downloading is that in progressive downloading the playback can start after a short buffering period, and there is no need to wait until the entire file is downloaded. In contrast to streaming, the file is saved in progressive downloading and next playback is a local playback that does not require a network connection.

2.3.2. Adaptive data transmission

PSS Release 5 [3GPP-5 22.233] already required that streaming service should provide a mechanism that adapts to network conditions, but did not actually describe any methods as to how to do this. Release 6 [3GPP-6 26.234] introduces several methods that can be used to adapt the service:

- Link characteristics
- Bitrate adaptation
- RTP retransmission

In the session setup, a client can send the original link characteristics (guaranteed bitrate, maximum bitrate and delay boundaries) to the server. With this information, a server can adapt the stream to fit the conditions.

Link characteristics may change over time and the values given at the beginning of the session may become outdated. Bitrate adaptation illustrated in Figure 4 allows the server to adapt the transmission bitrate to the available network bandwidth during an ongoing session.

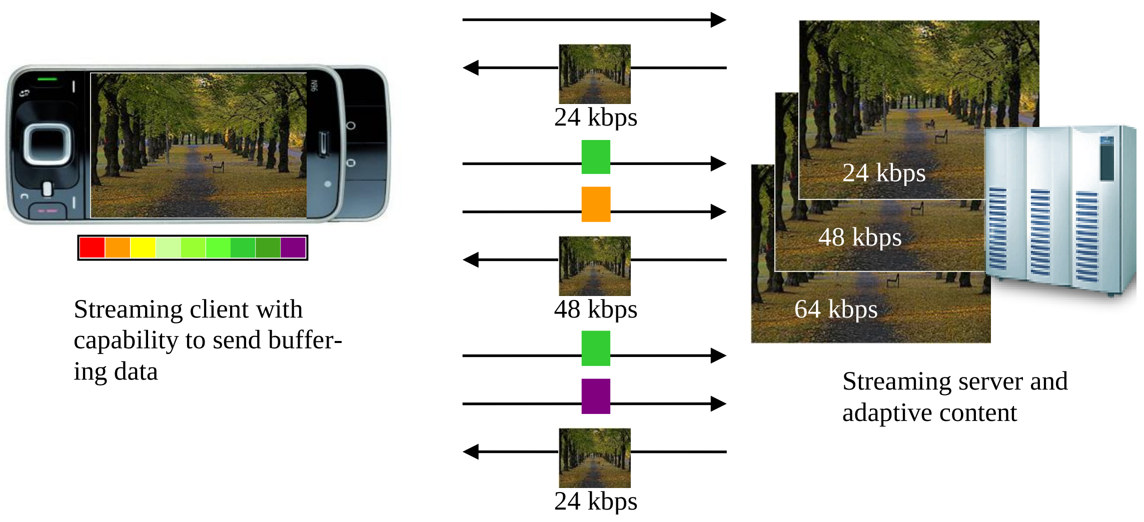


Figure 4: Adaptive streaming

In the session setup, the requested bitrate should be close to the available network bandwidth (or at least should not exceed the network bandwidth). Over time, the network bandwidth may reduce or increase, which may cause errors or insufficient network use. Bitrate adaptation attempts to adapt the transmission as close to the network bandwidth as possible. In the PSS session setup, a streaming client defines how much buffer space the server can utilize and defines the minimum target level of buffer that protects interruption-free playback. A server may utilize any means to maintain the buffer level. A streaming client will send all feedback information related to buffer status by using a RTCP protocol. A novel method described in Section 5.2.3 gives even more flexibility to adapt the stream. Due to the UDP based connection RTP packets can be lost. The typical reasons for this are low network quality (which causes single packet losses over time) and the network cell change (which causes a total block in dataflow during the change). RTP retransmission allows the client to request a server to resend lost packets. When a client notices that an RTP packet is lost, it will check whether the buffering status is such that a retransmitted packet can be used on time, and then the client sends an RTCP packet to request retransmission. A retransmitted packet should not prevent the sending of normal RTP packets; therefore, a server may decline the request if it hinders the normal transmission.

2.3.3. Secure data transmission

Even though a standard PSS client prevents a streaming session to be saved on a file, it is not difficult to implement a client that catches the media before playback and stores it to file (we had one for testing purposes to evaluate video quality). PSS Release 6 Annex K [3GPP-6 26.234] introduces methods that improve security of PSS.

Content encryption uses Open Mobile Alliance (OMA) Digital Rights Management (DRM), version 2 [OMADRM2] to create protected streaming services. A DRM protected RTP packet has a normal RTP header part and encrypted payload part. Acquiring the right to decrypt the streaming content follows the same rules as any OMA DRMv2 protected content. Release 6 also supports Secure RTP (SRTP) [RFC 3711] for integrity protection. SRTP offers both confidentiality and integrity protection, but PSS uses OMA DRM for confidentiality protection and SRTP only to integrity protection. With an SRTP master key, a client can ensure that the server is an authorized server and content provider.

2.3.4. Feedback mechanisms

Basic RTCP already provides a feedback mechanism that allows a client to report packet losses and delays. Extended RTCP, defined by PSS Release 6 [3GPP-6 26-234], allows the client to report the buffer status. In addition to these PSS Release 6 introduces a Quality of Experience (QoE) feedback method that allows reporting much wider data.

The QoE metrics allows the service to request quality metrics from the client. A client supporting the feature performs quality measurements, composes the measurements into QoE metrics and reports the metrics to the server. The PSS does not define how a server should use the information. What metrics are gathered and how often they are reported, are negotiated at the session setup. Gathered metrics include information about corrupted frames, buffering, packet losses and the playback frame rate deviation and jitter. The method is further described in Section 5.2.1.

2.4. FILE FORMATS

The basic set of codecs that need to be supported for a streaming service is defined in PSS Release 4. Almost all mandatory file formats were defined in Release 4. File formats are not studied in this thesis and no tests were performed to compare different media types or attempts to improve them. Therefore, this section only briefly describes the specified file formats of PSS.

2.4.1. Video

The basis of video streaming is H.263 [ITU-T H263] Profile 0, which is the only mandatory video codec of PSS. H.263 Profile 0 is the baseline of the codec specification, and it contains only mandatory functionality. PSS also includes another profile of H.263, which is Profile 3. This optional profile is designed to enhance coding efficiency and error resiliency of wireless devices. Profile 3 contains the following additional modes:

- **Advanced INTRA coding:** Improves the coding efficiency of INTRA macro blocks.
- **Deblocking filter:** Improves the coding performance and reduces the memory requirements.
- **Slice Structure mode:** Enhance the resynchronization of the stream after erroneous or lost data.
- **Modified Quantization:** Simplifies the encoding and improves the video quality.

Another optional video codec in PSS Release 4 is the MPEG-4 Visual Simple Profile [ISO 14496-2]. The Visual Simple Profile is designed for applications on mobile networks, where low bitrate and low resolution are typical.

A new optional video codec, H.264 (Also known as Advanced Video Codec (AVC)) [ITU-T H264] is introduced in PSS Release 6 [3GPP-6 26.234], H.264 is specially designed for lower bit-rate, being capable of providing better quality than previous codecs [Winkler 2006].

2.4.2. Audio

There are two mandatory audio codecs in PSS: Adaptive Multirate (AMR) [3GPP 26.071] and AMR-Wideband (AMR-WB) [3GPP 26.171]. These two codecs are designed for speech coding. Both AMR codecs include a set of bitrates that can be chosen depending on the available network bitrate. Table 2 summarizes these codecs. Both codecs can rapidly change the bitrate since each 20 ms frame can have its own bitrate. Section 5.2.3 reveals more information about bitrate change.

Table 2: AMR codec summary

Speech codec	**AMR**	**AMR-WB**
Bitrate	4.75 – 12.2 kbps	6.6 – 13.85 kbps
Sampling rate	8 kHz	16 kHz
Speech samples/frame	160	320

For more music types of audio, PSS release 4 uses Advanced Audio Coding Low-Complexity (AAC-LC) codec [ISO 14496-3] in mono and stereo. It is not a mandatory codec but should be supported, while AAC Long Term Prediction (AAC-LTP) is optional.

Scalable Polyphonic Musical Instrument Digital Interface (SP-MIDI) [SP-MIDI] for synthetic audio was added to PSS in Release 5 [3GPP-5 26.234]. SP-MIDI is designed for mobile applications and systems, and it allows different levels of polyphony (8-notes to 32-notes). A low-cost device can use 8-note polyphony and a high-end device uses 32-note polyphony, but they are both playing the same content from the server point of view.

More optional audio codecs are introduced in Release 6. AMR-WB+ [3GPP 26.290] and eAAC+ [3GPP 26.401]. AMR-WB+ improves the audio coding on lower bitrate speech, while eAAC+ improves the quality in higher bitrate music. In regard to synthetic audio, Release 6 introduces Mobile Downloadable Sounds (Mobile DLS) [MMA-MDLS] and its file format Mobile eXtensible Music Format (Mobile XMF) [MMA-MXMF]. Mobile DLS is a simplified version of DLS Level 2 and the minimum device requirements are more relaxed to suit mobile devices. A Mobile XMF file format can include both SP-MIDI and Mobile-DLS sounds.

2.4.3. Other media types

In addition to continuous video and audio, PSS supports two discrete media groups:

- Images, bitmaps and graphics
- Text and scenes

Only mandatory image format, Joint Photographic Experts Group (JPEG) [ITU-T T.81] and its JPEG File Interchange Format (JFIF) are specified in PSS Release 4, and the support is limited to baseline and progressive formats. Support of Graphic Interchange Format (GIF) [GIF 87a & GIF 89a] bitmap format versions is recommended in PSS 4. A Portable Network Graphics (PNG) [RFC 2083] was added in Release 5 to improve bitmap offering. Vector graphic support is mandatory in PSS Release 5 with Scalable Vector Graphics (SVG) Tiny

[W3C SVG-Tiny]. SVG-Tiny can be used in restricted mobile devices, while SVG Basic can be used optionally in high end mobile devices.

Mandatory Synchronized Multimedia Integration Language (SMIL) [W3C SMIL] presentation and scene support is included in PSS Release 4. SMIL is based on eXtensible Hyper Text Markup Language (XHTML) Mobile profile [XHTML]. An optional Timed-text is included in PSS release 5, but this is used only in downloading and not with streaming. Timed-text allows adding subtitles to content. PSS Release 7 adds Dynamic and Interactive Multimedia Scenes (DIMS) Mobile profile [3GPP-7 26.142], which offers an additional scene description method to SMIL

Chapter 3

Packet-switched networks

MOBILE networks are divided into two systems: circuit-switched and packet-switched. Circuit-switched networks are traditionally used for voice calls, while packet-switched networks are used for data transmission. This chapter describes the characteristics of a packet-switched network from a streaming point of view. The purpose is not to describe the networks in detail, but only the features that have an impact on the PSS.

3.1. NETWORK OVERVIEW

The basic elements of the Packet-switched (PS) network are illustrated in Figure 5. The names may vary according to the technology used, but the basic structure is always the same. The names seen in Figure 5 are used in GPRS networks.

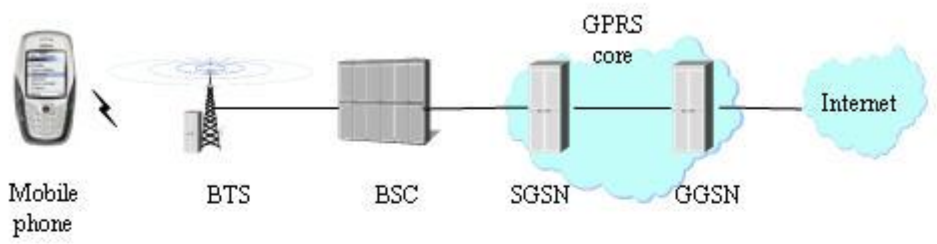


Figure 5: Basic Elements of PS network

A mobile phone connects to the network Base Transceiver Station (BTS), which handles radio signal transmission and encryption. The BTS is connected to the Base Station Controller (BSC), which allocates resources between several BTSs. Typically one BSC is connected to tens of BTSs. The BSC communicates with the Serving GPRS Support Node (SGSN), which is responsible for the delivery of the data packets. The SGSN also handles routing, mobility management, authentication and charging, ciphering and data compression functions. The final element of the PS network, The Gateway GPRS Support Node (GGSN), functions between the GPRS network and other PS networks such as the Internet. The GGSN functions as a router to the sub-network, since it hides the GPRS infrastructure from the external network.

Packet-switched (PS) networks are divided into different characteristics, which are designed for certain types of service. For simplicity, we divided the services into two units: continuous and discrete. Continuous service means that there is continuous data flow from the service provider to the user and streaming is one example of such service. Discrete service means that the data flow is random, and there can be periods when no data is sent. A typical example of a discrete service is browsing a webpage.

Each PS network can handle the discrete services, since there are no time constraints and the TCP handles the potential packet losses. Due to limited network bandwidth, service can be designed for mobile use, which typically means that the webpage is smaller and data objects are small.

Continuous services are much more demanding for PS. Each data packet needs to arrive within certain time limits and some standards use UDP, which does not guarantee a lossless connection. Also the available network bandwidth may vary over time. Table 3 shows how networks can be divided into bitrate guaranteed and non-bitrate-guaranteed segments for continuous services. GPRS does not have traffic classes, but every service is treated equally with the characteristics similar to the Background traffic class. The EGPRS network has different traffic classes. Interactive and Background traffic classes divides the traffic into four priority groups, in which Background has the lowest priority and the Interactive class with traffic handling priority 1 has the highest priority. EGPRS has Streaming and Conversational classes, but the EGPRS could not guarantee available bitrate in the first releases. In GERAN 5 release, EGPRS begins to use the same core network as WCDMA, and provides the same characteristics. In the first WCDMA release, Streaming and Conversation classes can guarantee an available network bandwidth and traffic delay boundaries.

Table 3: Bitrate guaranteed mobile networks for continuous services

Traffic class	**GPRS**	**EGPRS**	**WCDMA**
Background	No	No	No
Interactive	N/A	No	No
Streaming	N/A	No (Yes after GERAN 5)	Yes
Conversational	N/A	No (Yes after GERAN 5)	Yes

One may wonder why this study is limited to non-bitrate-guaranteed networks and does not study bitrate guaranteed networks at all. First, non-bitrate-guaranteed networks are much more widely available than bitrate guaranteed networks. Second, there is very little to study in bit-rate guaranteed networks, since these network have been designed previously for continuous service. The purpose of this study was to take the existing network and determine what could be done without requiring changes to the network.

If we look at what kind of mobile phones was sold for example by Nokia on the European market (Figure 6 shows the situation November 2010), 13 percent of these devices are not equipped with a packet-switched connection and 3 percent of the devices have a GPRS connection only. An EGPRS connection can be found in 25 percent of the devices and 58 percent have a WCDMA connection. Thus there is enormous potential for PSS use in regard to phones.

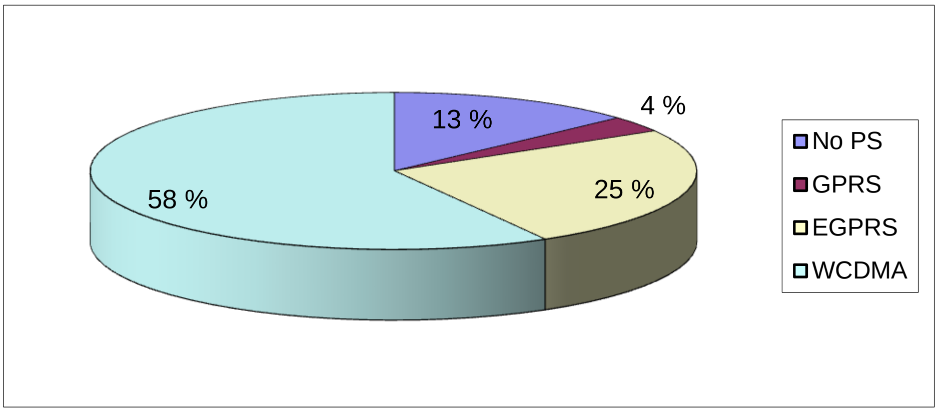


Figure 6: Data Transfer technologies in Nokia phones sold in Europe

3.1.1. GPRS

A General Packet Radio Service (GPRS) was a set of new GSM bearer services that provided packet mode transmission. The first version of GPRS is called a GPRS Release '97. Based on the specification [GSM 02.60 Rel 97], GPRS is designed for applications with the following characteristics:

- Intermittent data transmission
- Frequent transmission of small volumes of data
- Infrequent transmission of large volumes of data

GPRS can also be used in more demanding services, but this is optional from a specification point of view. GPRS does not provide any means of guaranteeing data flow. If signal quality is weak, a more reliable Coding Scheme (CS) can be used that reduces the bandwidth. If there are too many users on the network, the system can free TimeSlots (TS) from other users, which again reduces the bandwidth. Table 4 shows the theoretical bitrates of GPRS in different CS/TS combinations [GSM 03.64 Rel 97]. Four timeslots are typically commercially available (i.e. one phone cannot receive more than four timeslots). Some of the latest phones can receive more than four timeslots, but this is still rare. Also Coding Schemes 3 and 4 require such a strong signal quality that it may be hard to receive. [Dogan 2002] illustrates the quality of different CS levels, and it is easy to see that CS3 is unable to achieve acceptable video quality. More details on achieved bitrates can be found in Section 4.2.

Table 4: Theoretical GPRS bitrates

	1 TS	**2 TS**	**3 TS**	**4 TS**	**5 TS**	**6 TS**	**7 TS**	**8 TS**
CS-1	9.05	18.1	27.15	36.2	45.25	54.3	63.35	72.4
CS-2	13.4	26.8	40.2	53.6	67.0	80.4	93.8	107.2
CS-3	15.6	31.2	46.8	62.4	78.0	93.6	109.2	124.8
CS-4	21.4	42.8	64.2	85.6	107.0	128.4	149.8	171.2

The cell reselection behavior of GPRS is different than in GSM. In GPRS, the device is the master that decides when the cell reselection is needed. This device measures the signal strengths of the serving cell and neighbor cells. Based on this information, the device decides the moment of cell reselection. The complexity of the cell reselection depends on the level of change. Cell reselection can be divided into five categories:

1. Inter Cell CR (simplest): the cell is changed to another cell within the same BSC.
2. Inter BSC CR: Change to the cell served by another BSC.
3. Inter SGSN CR: Change to the cell served by another SGSN.
4. Inter GGSN CR: Change to the cell served by another GGSN.
5. Roaming (hardest): Change to the cell served by another operator.

The cell reselection criteria are improved in GPRS compared to GSM, but the device-based decision making is unfortunately slower than the network-based decision. GPRS cell reselection times and other results related to the GPRS can be found in Section 4.2.

3.1.2. EGPRS

Enhanced GPRS (EGPRS) is a component of the Enhanced Data encoding for Global Evolution (EDGE). EDGE includes Enhanced Circuit Switched Data (ECSD) and EGPRS components. This thesis studies only the EGPRS component. Some may also refer to first EGPRS release as a term of GPRS Release '99.

Compared to GPRS, Enhanced GPRS provides much higher data rates and some improved error handling methods. EGPRS also has some benefits over WCDMA. EGPRS network cell size is greater compared to WCDMA network cell size. EGPRS can use an existing GSM base station network, while WCDMA would require a denser network. The cost of building extensive WCDMA coverage is much higher than the cost of extensive EGPRS coverage. This cost may very well prevent the usage of WCDMA outside densely populated areas.

EGPRS [GSM 03.64 Rel 99] defines 9 different Modulation and Coding Schemes (MCS). The best protection is provided by MCS-1 with 8.8 kbps per timeslot. MCS-9 provides the lowest protection and is able to achieve 59.2 kbps per timeslot. Table 5 shows the bitrates for each timeslot. Typical EGPRS phones can access 4 timeslots, but the best bitrate values may not be error free (See Section 4.2)

Table 5: Theoretical EGPRS bitrates

	1 TS	**2 TS**	**3 TS**	**4 TS**	**5 TS**	**6 TS**	**7 TS**	**8 TS**
MCS-1	8.80	17.6	26.4	35.2	44.0	52.8	61.6	70.4
MCS-2	11.2	22.4	33.6	44.8	56.0	67.2	78.4	89.6
MCS-3	14.8	29.6	44.4	59.2	74.0	88.8	103.6	118.4
MCS-4	17.6	35.2	52.8	70.4	88.0	105.6	123.2	140.8
MCS-5	22.4	44.8	67.2	89.6	112.0	134.4	156.8	179.2
MCS-6	29.6	59.2	88.8	118.4	148.0	177.6	207.2	236.8
MCS-7	44.8	89.6	134.4	179.2	224.0	268.8	313.6	358.4
MCS-8	54.4	108.8	163.2	217.6	272.0	326.4	380.8	435.2
MCS-9	59.2	118.4	177.6	236.8	296.0	355.2	414.4	473.6

EGPRS provides a few new improvements related to quality. Incremental redundancy [GSM 04.60 Rel 99] is an enhancement for the acknowledged Radio Link Control (RLC) mode. In traditional acknowledged mode; already used in GPRS; the RLC block is retransmitted if an error occurs. In incremental redundancy, this retransmission is enhanced by keeping the erroneous block and retransmitting the block with lower bitrate MCS, which offers better error protection. Even if the retransmitted block still contains errors, it is possible to combine the original and retransmitted block and check if the combined block is error free. If there are still errors, the new retransmission can be requested. It is possible to continue retransmission as long as needed. Naturally, retransmissions reduce the bandwidth reserved for normal data transmissions and causes extra delay, preventing its use in all real-time services, but it can be used in streaming services since application buffers hide the delays from the end-user.

A Link Adaptation [GSM 05.09 Rel 99] algorithm allows selecting the optimum MCS that matches the reliability QoS requirements, depending on radio link conditions. This implies that the performance in terms of throughput, delay and error ratio varies continuously. The throughput depends on the link quality in terms of carrier/interference (C/I) values. This algorithm aims to predict the future C/I and based on this estimate, it decides what the correct MCS is.

EGPRS uses the same cell reselection mechanism as GPRS, but a Network Assisted Cell Change (NACC) was introduced in the GSM EDGE Radio Access Network (GERAN) Release 4 [GSM 44.060 Rel 4], which is the next release of EGPRS. NACC improves the cell reselection behavior, since GPRS cell change was not originally designed for services that have real-time requirements and continuous data flow. In NACC, the network can send cell information in advance to a mobile device, and the device can use this information when making a cell reselection. This decreases the cell reselection time from seconds to a few hundreds of milliseconds. NACC can be done with cells that are within the same Base Station Controller (BSC).

Another improvement in GERAN 4 is the Extended Uplink Temporary Block Flow (TBF) [GSM 44.060 Rel 4]. Radio interface protocols are initially designed to free the unused radio resources as soon as possible. If traffic is bursty, this may lead to frequent resource set-ups and release. Unnecessary TBF set-up and release can be avoided by delaying the release of TBF. GPRS Rel'97 already had the delayed TBF from the downlink side, but GERAN Rel'4 introduced the same to the uplink. The impact on PSS is minor, since most of the traffic is in the downlink direction, but it may improve some control and feedback message passing times.

Before GERAN Rel' 5 the circuit-switched and packet-switched core networks were two separate networks. In GERAN Rel'5, the GERAN and UMTS Terrestrial Radio Access Network (UTRAN) uses the same interface and common core network called an Internet Multimedia Subsystem (IMS) [GSM 23.228 Rel 5]. It brings many changes to GERAN, but not all of these have an impact on PSS. In the following section, we describe those that may improve streaming services.

Harmonization of GERAN and UTRAN core networks allows GERAN to connect with the 3G core network and enables GERAN to provide the same set of services as UTRAN. This means that GERAN Streaming and Conversational traffic classes can guarantee bitrates and delay bounds.

The header compression reduces the overhead that different headers cause in each packet. The minimum RTP/UDP/IP header is 40 bytes (RTP 12 bytes, UDP 8 bytes and IPv4 20 bytes) and the bandwidth used by headers was 10-15 % depending on the bitrate (a higher bitrate also renders the packet size larger, which reduces the header overhead). The header compression can slightly increase the bandwidth available for media.

GERAN 5 expands the Network Assisted Cell Change (NACC) to also cover cell reselections between Base Station Controllers (BSC) and between GERAN and UTRAN cells. The goal was to reduce the cell reselection times.

The IMS framework, in the UMTS core network, was further developed in GERAN Rel' 6. Due to the fact that the GERAN 5 IMS was too rigid and complicated to specify, GERAN 6 introduces a Flexible Layer One (FLO) principle [GSM 45.092 Rel6]. Rather than having fixed coding schemes in specifications, FLO provides a framework that allows the coding scheme to be specified and optimized at the session setup. FLO does not directly improve existing PSS, but it allows more flexibility to future improvements and makes it easier to introduce them.

Another improvement of GERAN 6 is the Single Antenna Interference Cancellation (SAIC) [GSM 45.903 Rel6]. The main idea of SAIC is to cancel or suppress co-channel interference coming from other cells transmitting in the same frequency. From a PSS point of view this can improve the network bandwidth, since higher MCS can be used.

Multimedia Broadcast Multicast service (MBMS) [GSM 22.146 Rel 6] is a new service introduced in GERAN 6. MBMS allows sending the data to a common channel and air interface and not sending the same data many times. MBMS is a different kind of service than PSS, but the technique could be used in live streaming, which is a more broadcast/multicast type of service.

Some minor enhancement to improve Streaming traffic class cell reselection handling is also included in GERAN 6, aiming to reduce packet losses during cell change.

3.1.3. WCDMA

Wideband Code Division Multiple Access (WCDMA) technology is a third-generation mobile technology. Another name used for it is UMTS Terrestrial Radio Access Network (UTRAN). One of the major differences between the 2G and 3G is the means by which the cells are differentiated from each other. GPRS and EGPRS use Frequency Division Multiple Access (FDMA), in which each cell has a different frequency. It is further enhanced with Time Division Multiple Access (TDMA), whereby each frequency is divided into a timeslot and each user is allocated to one or more timeslots. Changing from one cell to another cell means that a mobile device must change the radio frequency, which causes a break in dataflow (i.e. there is a period when the connection to old cell is no longer maintained, but the connection to new cell cannot be established, since device has to first change the radio frequency). In CDMA, cells are differentiated with a pseudo-random code that allows them to use the same frequency. This allows, for example, soft cell changes without breaks in dataflow. In this thesis, WCDMA technology is not described in detail, but the features that have impact on PSS are introduced.

WCDMA Release 99 is the first version of the technology [GSM 22.100 Rel 99 & GSM 25.201 Rel 99]. Although the 3G requires changes in both CS and PS networks, they are still two separate networks. Interoperability between GSM and UMTS must be guaranteed and inter-system handovers must be possible in both directions. The WCDMA Release 99 uses the same QoS profile as EGPRS release 99, but WCDMA is able to guarantee bitrates and delay bounds. Maximum bitrate in the network is 2Mbps, but in practice, the first phones allowed only a 384-kbps connection. A WCDMA network has four different types of cell changes. Intra-frequency handover, which occurs within the same frequency, is divided into two parts: soft handover means cell change to another cell that works with the same frequency, whereas softer handover means changing the sector within the same cell. Depending on network planning, there might be locations where soft handovers are not possible and inter-frequency handover is needed. When entering or exiting a 3G-network coverage area, an inter-system handover between GSM and WCDMA is required.

Later UTRAN releases do not have significant impact on PSS. UTRAN Release 4 does not have any major updates from a PSS point of view. Most of the changes in Release 4 are related to creating more flexibility and cost efficiency to the core network, for example, the bearer independent core network [GSM 23.205 Rel 4]. UTRAN Release 5 changes only the PS component of the network and leaves the CS untouched. Release 5 introduces two major elements. The first is the IMS that was described in the previous section, since in this release both, WCDMA and EGPRS, use a common core network. The second is upgrading the network to support real-time voice and the other delay-sensitive services [GSM 25.933 Rel 5]. UTRAN Release 6 includes MBMS such as GERAN Release 6.

3.2. QUALITY OF SERVICE

The Quality of Service profile defines the expected quality terms of the service. The first profile was introduced in GPRS (Rel' 97) and it was further developed with Release 99.

3.2.1. QoS Profile of Release 97

GPRS offers a means to describe the characteristics of needed connections by using Quality of Service (QoS) parameters. Table 6 shows the overview of QoS parameters. More details can be found in [GSM 02.60 Rel 97 & GSM 03.60 Rel 97]

Table 6: QoS parameters in GPRS

QoS parameters	Description
Precedence class	Importance of the service
Delay class	Maximum delay of the packets
Reliability class	Probability of errors
Peak throughput class	Maximum data rate
Mean throughput class	Mean data rate

Theoretically PSS demands service that needs the best QoS parameters. It is easy to define the best value in Precedence and Delay classes, and throughput classes depend on network and media bandwidth. But the Reliability class is difficult to define, since it combines error protection and real-time demands (more error protection causes more delays).

The *Precedence class* indicates the relative priority of maintaining the service. Under abnormal conditions the network may be unable to maintain all services and with the Precedence class, the network knows what services are dropped if such conditions occur (low Precedence class services are dropped if high Precedence class service requires a network resources) From the streaming point, Precedence class 1 should be used. More details of Precedence class can be found from Chapter III B of [P2]

The *Delay class* defines the upper bounds of the end-to-end delay. The parameter does not guarantee any delay boundary, but it states that the network should allocate adequate transmission resources to support the given delay values. From a streaming point, Delay class 1 should be used. Exact Delay class values can be found from Table 3 of [P2]

The *Reliability class* defines the probability of data loss (including out of sequence and duplicate data delivery, and data corruption). Reliability class 1 is highly non-transparent (i.e. GPRS Tunneling Protocol (GTP), Logical Link Control (LLC) and Radio Link Control (RLC) layers work in the acknowledged mode, which retransmits erroneous data) providing the best reliability. On the other hand, reliability class 5 is fully transparent in passing through all the errors to the upper layer. Reliability classes 4 and 5 are used for real-time traffic such as streaming. Table 2 of [P2] illustrates the differences between each Reliability class.

The *Peak Throughput class* and *Mean Throughput class* indicates the maximum and average bitrate of the requested content from the server. Since the theoretical maximum bitrate of the GPRS network is 171.2 kbps, the Peak Throughput class 6 is the highest that could be used and most probably class 5 is the highest needed. From a PSS point of view, Mean Throughput class values are extremely low and only classes 16-18 (22 kbps-111 kbps) could be used. Tab les 4 and 5 of [P2] shows the full list of different bitrates.

Section 4.2 describes what type of PSS experience was gained with these values, and Section 5.1 shows the proposed changes to these parameters.

3.2.2. QoS Profile of Release 99

Release 99 introduces new Quality of Service parameters [3GPP 23.107 Rel 99]. These parameters are used in both EGPRS and Universal Mobile telecommunication System (UMTS) networks. Some of the parameters are for UMTS only since EGPRS cannot utilize them. Parameters are divided into four QoS classes. Table 7 summarizes the difference of the classes.

Table 7: Release 99 QoS classes

Conversational class	Real-time services such as Voice over IP and Video conferencing. Strict delay bounds, lower error ratio demands and possibility of guaranteed bitrate.
Streaming class	Near Real-time services such as Video on Demand. Relatively strict delay bounds, lower error ratio demands and possibility of guaranteed bitrate.
Interactive class	Request-Response type of services such as web browsing. No delay bounds, stricter error ratio demands, no guaranteed bitrate, possible to prioritize the traffic within class.
Background class	Services that have no time limits such as background downloads. No delay bounds, stricter error ratio demands, no guaranteed bitrate, lowest priority of all traffic.

QoS parameters, possible values and brief explanations are provided in Table 8. SDU means Service Data Unit and the BER is the Bit Error Rate. The Release 99 QoS profile contains much more exact number values compared to Release 97, which contained only parameter classes.

Table 8: Release 99 QoS parameters

Parameter	**Value**	**Description**
Maximum bitrate	< 2 048 kbps	Upper limit application can handle
Delivery order	Yes/No	Should SDUs be delivered in order (i.e. are out-of-sequence SDUs accepted)?
Maximum SDU size	1 500/1 502 octects	Maximum allowed SDU size for admission control and policy
SDU format information	Size in bits	List of possible exact sizes of SDU for transparent RLC protocol mode: makes the bearer less expensive.
Delivery of erroneous SDU	Yes/No	Should erroneous SDU be delivered or discarded?
Residual BER	$5x10^{-2}$ – $6x10^{-8}$	Undetected bit error rate
SDU error ratio	10^{-2} - 10^{-6}	Fraction of SDUs lost or detected as erroneous
Transfer delay	100 ms or 280 ms	Tolerated delay (Conv. and Stream. class only)
Guaranteed bitrate	< 2 048 kbps	Bitrate guaranteed by the UMTS bearer (Conv. and Stream. class only)
Traffic handling priority	1,2,3	Relative importance of SDUs (Interact. class only)
Allocation/Retention priority	1,2,3	Relative importance compared to other UMTS bearers (unlike other parameters, a mobile terminal cannot negotiate this value)

Bitrate values are for UMTS networks only. EGPRS cannot reach such values. Also EGPRS cannot guarantee bitrates before GERAN 5. Thus, even if a guaranteed bitrate is requested, it cannot be granted. Also the Transfer Delay bounds cannot be guaranteed in EGPRS before GERAN 5.

The specification [3GPP 23.107 Rel 99] also includes information on how to map values between GPRS and EGPRS/UMTS QoS parameters. This mapping can be used if service is handed over to another network.

3.3. STREAMING RELATED FEATURES

There are numerous features in each network technology that have no impact on the streaming service, but are related to, for example, traditional voice calls. This section lists the basic features related to streaming service and also the greatest problems that the current networks have.

3.3.1. Bandwidth

Network bandwidth is directly linked to media quality. Since streamed data is not stored on the device, the device can only show data that has come through at a given time. A streaming client has a small buffer to cover errors, but its capacity is limited. Figure 7 shows the quality differences between typical GPRS, EGPRS and WCDMA bitrates and compares these to the original sequence.

Figure 7: GPRS,EGPRS, WCDMA and Original video quality

Figure7A depicts the GPRS stream (30 kbps video bitrate). The stream has lost almost all the details and the edges are not sharp. Figure 7B depicts an EGPRS stream (84 kbps video bitrate). The stream is much sharper and all the windows of the building are visible. Also the edges of the car are more visible. Figure 7C depicts the WCDMA stream (330 kbps video bitrate) and it is very difficult to notice a difference between it and the original stream (Figure 7D). Even the brick structure of the building can be seen. The next enlarged figures highlight the difference even more. Figure 8A is very blurry, while Figure 8B begins to improve. Even with the enlarged figures, it is very difficult to see any quality difference between Figure 8C and Figure 8D.

A B C D

Figure 8: GPRS, EGPRS, WCDMA and Original video quality (enlarged)

Bandwidth may also vary a great deal during the streaming session. The change can be positive - increased bandwidth or negative – decreased bandwidth. Section 5.2.3 describes the studies and solution to solve bandwidth variation related problems.

3.3.2. Cell change

Most of the time, the cell change causes a gap in the dataflow. Figure 9 shows what happens to the stream during cell reselection. The sent stream is the original stream with all the frames and Intra-frames (marked as I). Due to the nature of video compression, only the Intra-frames are full frames. The frames between I-frames only include the difference between it and the previous frame; therefore, in order to decode the full frame to the screen, the decoder needs more than one frame. In this scenario, cell reselection occurs after the 8th frame, causing frames 9 to 11 to disappear. Since frame 12 would need previous frames to be fully decoded, a prediction error occurs that lasts until the next Intra-frame occurs. Severity of the prediction error depends on the motion of the stream. If there is hardly any motion in the scene, the prediction error can be almost not visible, but with high motion the error is very significant. The length of the prediction error depends on the Intra-frame rate.

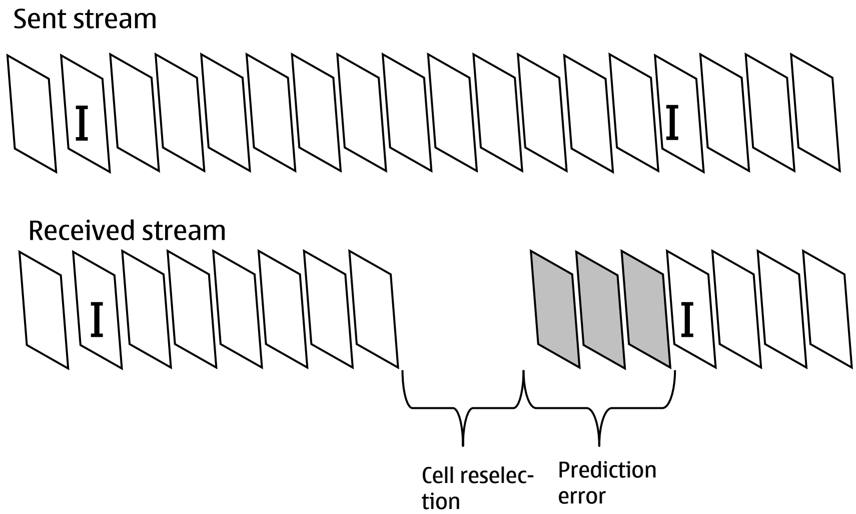


Figure 9: Cell reselection errors

With QoS parameters, it is possible to request a connection that buffers the data during cell reselection (e.g. GPRS Reliability classes 1 and 2). In this case, the frames that the network may have sent during cell reselection are stored and sent after the cell reselection. If the streaming client has a large enough buffer, it is possible that cell reselection occurs without errors. The only impact is that the client's data buffer level lowers, and if cell reselections occur often, it eventually dries the buffer out and causes reloading of the buffer, which the user sees as a pause in playback. Section 5.2.2 describes the studies and solutions to solve the cell reselection related problems.

3.3.3. Data losses

In addition to cell changes, data losses can occur during the entire streaming session. During the transfer, an RTP packet may become erroneous. Depending on what part of the packet error affects, the packet can be discarded by the decoder or it may never arrive to the streaming client. Regardless of the reason why the packet is not decoded, it will create a similar kind of error as described in the cell change. Even though only a single frame is lost, there may still be a long prediction error period. If the buffer on the client side is large enough, the client can request retransmission of the lost packet (PSS Release 6 feature), but with a small buffer, the playback time occurs before the retransmitted packet arrives. Section 5.2.1 describes studies on data loss.

3.3.4. Transfer delay

Bandwidth variations on the network cause additional delays in packet arrival. It is possible that this additional delay may even cause the packets to arrive in the wrong order. A large enough client buffer can handle the delay jitter and reorder the packets. Section 4.2 shows a proposal of the buffer sizes.

Chapter 4

Characteristics of PSS

PREVIOUS chapters described the basic features of a Packet-switched Streaming Service and Packet-Switched networks. This chapter describes the central findings of the study and shows the characteristics of the service in a basic packet-switched network. It also compares the results with other studies.

4.1. TESTING ENVIRONMENT

The testing environment consisted of two laptops, one mobile phone and a mobile network. The first laptop included a streaming server connected to the Internet with a LAN cable. The second laptop included a streaming client (a player) and it was connected to the mobile device with infrared or cable (cable was used whenever possible since it is considered the most reliable connection type. Also Bluetooth could have been used but it was not available at the time). A mobile device made a packet-switched connection to the mobile network under testing (GPRS, EGPRS or WCDMA network). Figure 10 illustrates the environment.

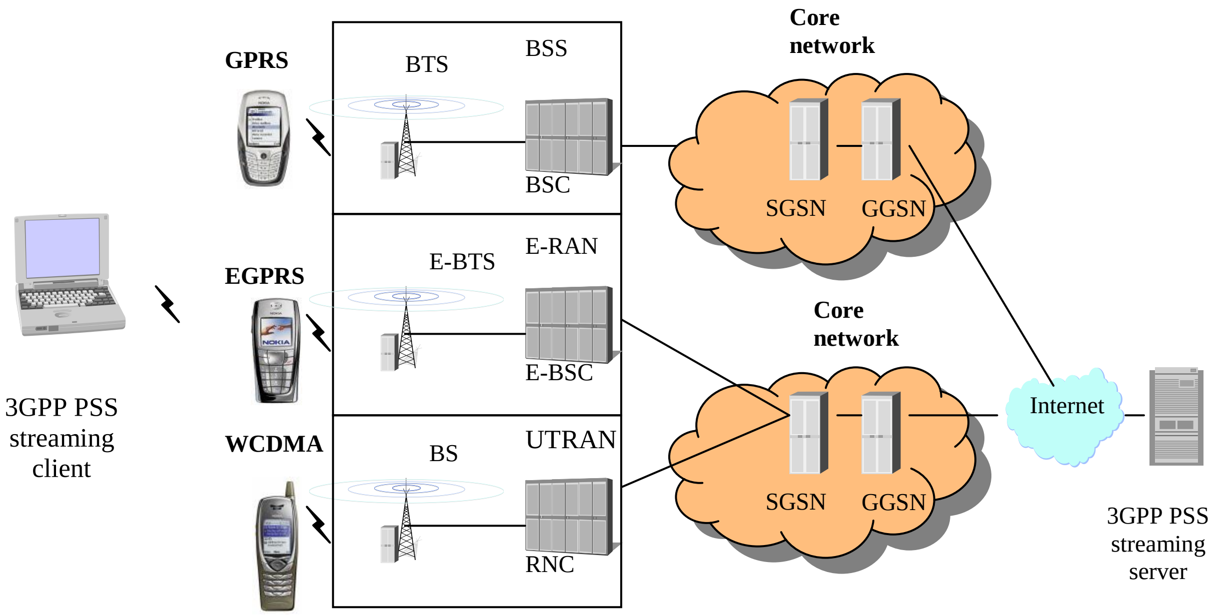


Figure 10: Testing Environment

The mobile network tested was a Nokia companywide test network. It has some load, since it is used in all types of tests, but the load may differ from the commercial network (i.e. it is not a closed, lab room, environment, but a real network with multiple cells and several hundred meter coverage). It also allows manipulating the network. For example, cell reselections were caused by adding noise to the cell where a mobile device was connected. Also we were able to select the used Timeslots and Coding Schemes, but otherwise we didn't change the network parameters (test network is mainly used by testing device software before commercial release, so the setup is as close as it can be of commercial environment). The conditions of the test environment were kept as similar as possible. Same PCs were used throughout all tests and also the streaming client and server applications were always the same (excluding the cases where some improvement was under testing). Connectivity of mobile device was kept the same through the test session (but may have varied between different phones)

The reason for using a mobile device only as a modem and not a streaming client itself was due to the heavy log file gathering operations. The problem was that the capacity of the mobile device is unable to handle the streaming playback properly if the device must write many log files simultaneously; consequently the test results would have been biased. Especially since raw video files, stored during streaming, required a great deal of resources even from the laptops, adding this kind of data gathering on a mobile device in addition to streaming would have changed the test results.

Since the tests were performed in the very early stages of evolution of the PS network, the EGPRS and WCDMA networks contained only Release 99 features. The streaming server and client were implemented according to PSS Release 4 specifications. Since PSS Release 6 was under specification, some of these features were added later to the PSS client and server (See Section 5.2.2 and 5.2.3 on cell reselection management and bandwidth adaptation).

The content used in the test was a 60-second movie trailer, with fast motion and several scene jump cuts. An AMR speech codec was used for audio and H.263 to encode the video (the codecs used are irrelevant in these tests, since we were analyzing the behaviors of the network and not the suitability of the codecs). One could have used an H.264 video and AAC audio in these tests, but they were not available at the time. The video codec used is irrelevant to the test results, since the measurements are done at the RTP packet level and the results would have been the same for all codecs. H.264 provides better quality than H.263 and requires less bandwidth (to achieve the same quality), but it is unable to handle data losses any better than H.263 (at the RTP level, although it has better error resilience on codec level). Bitrates were adjusted according to the network limits. The packetization of the media was the same as in IETF Specifications [RFC 3267, RFC 2429]. The speech packets had constant sizes depending on the bitrate used, while video packets varied. The video packetizing strategy was to include one frame per packet, with a limit of 1430 bytes per packet (including header). If the frame was too large to fit into one packet, it was divided into several packets. Table 8 of [P2] summarizes the encoding parameters.

While testing, we monitored the following metrics (all the values were logged by our streaming client):

- **RTSP signaling delay (message sending and receiving times in milliseconds):** There are three types of signaling (session setup, session change and session teardown) that were monitored. Session setup is the heaviest signaling operation since it includes a TCP handshake and several rounds (request-response pairs) of RTSP messages. A session change and teardown only contains one round of RTSP messages.
- **Bitrate (cumulative size of received packets/time of sequence):** True bitrate of the network taking into account the error protection needed for service.
- **Packet loss rate (number of lost packets/number of sent packets):** Packet loss rate of the network without additional causes of loss (i.e. cell reselection or network overload)
- **Cell reselection time (the gap of RTP packet transmission in milliseconds):** The time when no packets are sent from network to the device due to the cell reselection.
- **Buffering requirement (client parameter in milliseconds):** The size of the buffer required to handle the packet arrival fluctuation and cell reselection.

4.2. PSS IN MOBILE NETWORKS

The tests were done in three different mobile networks (GPRS, EGPRS and WCDMA). This section provides all the results obtained. The tests were done with QoS values proposed in the specifications (e.g. Reliability class 4 or equivalent). More information on the GPRS tests can be found in [P1, P2]. EGPRS tests are described in detail in [P3] and WCDMA tests in [P4].

4.2.1. RTSP signaling

The connection set-up initiates the entire streaming session, which includes a TCP synchronization and initial RTSP signaling (RTSP Describe, Setups and Play requests and their responses). Table 9 shows the delay in various networks.

Table 9: Connection setup delays

Connection setup	**WCDMA**	**EGPRS 2+1**	**GPRS 3+1**
Average (s)	1.4	5.2	7.3
Minimum (s)	1.2	5.1	5.3
Maximum (s)	1.9	5.3	8.8
95^{th}-percentile (s)	1.8	5.3	8.8
Test runs (n)	10	10	10

The results show that on average, WCDMA is 3.7 times faster than EGPRS and 5.2 times faster than GPRS. To understand the difference between theory and empirical results, GPRS results are compared. The total size of the RTSP messages is around 1250 bytes (925 bytes are transferred to downlink and 325 bytes to uplink direction). The theoretical bitrate of GPRS 3+1 CS2 connection is 40.2 kbps downlink and 13.4 kbps uplink. If we assume that 85% of this bitrate is available to RTSP messages (see Table 11 for media bitrates) then 7400 bits are transferred with 34.2 kbps connection to downlink and 2600 bits with 11.4 kbps to uplink. In theory it takes 0.22 seconds to transfer the needed data to downlink and 0.23 second to uplink, which yield the total time to 0.45 seconds. The total signaling time (7.3 seconds) includes also TCP synchronization which took on average 2.4 seconds. This indicates that on average 4.5 seconds (60% of the total time) is used by streaming server, streaming client and various GPRS network components to something else than pure transmission of the data (e.g. parsing, routing, loss recovery).

During the session, or at the teardown of the session, only one RTSP Request-response pair is sent between streaming client and server. Table 10 shows a Pause message as an example of such a single pair.

Table 10: Single RTSP message pair delay

Signaling delays	**WCDMA**	**EGPRS 2+1**	**GPRS 3+1**
Average (s)	0.4	1.1	1.6
Minimum (s)	0.3	0.7	1.2
Maximum (s)	0.5	1.6	2.0
95^{th}-percentile (s)	0.5	1.5	1.9
Test runs (n)	10	10	10

WCDMA is 2.8 times faster than EGPRS and 4.0 times faster than GPRS. It is self-evident that increased bandwidth makes the signaling faster, but the actual bandwidth increase of the WCDMA (384 kbps) compared to GPRS (40.2 kbps with three timeslots) is 9.5 times, which indicates that the benefit of the additional bandwidth has reached its limits and the processing of the messages becomes more apparent in the delay. These values can also be considered as reaction times of the streaming system. In Chapter 5, many methods are based on a client sending a request to a server, and with these values it can be estimated that, for example, in GPRS, it takes more than 1 second before a client can expect to receive the requested result.

4.2.2. Media bitrate

Theoretical bitrates were described in Section 3.1. A true bitrate, however, depends on error correction, load, and device capabilities. When calculating the media bitrate (i.e. the bitrate an end-user can experience in streaming) also the header overheads need to be taken into account. In the tests, the following media bitrates were achieved (Table 11). Note the timeslots used.

Table 11: Media bitrate values

Network bitrate	***Media bitrate***
GPRS 3+1 (CS-2)	35 kbps
EGPRS 2+1 (MCS-7)	80 kbps
WCDMA	342 kbps

Commercial networks may have a significantly larger load than in the test networks used in the tests. In [Beaufort 2002], the commercial T-Mobile GPRS networks were able to yield only a 22-kbps bitrate, which indicates that the test network used in this study was lightly loaded. [Nguyen 2004] claims that only EGPRS MCS-1 can be used to achieve good video quality, which would allow only 19 kbps with two timeslots (Compared to 89 kbps with MCS-7). They base their claim on the fact that the Signal-to-Noise (SNR) ratio should be above 10 dB. The basic assumption of SNR is probably correct but they did not provide enough information about the nature of the errors causing losses. Also [Kodikara 2002] successfully used MCS-5 in the link adaption tests, which would seem to indicate that Nguyen's claim [Nguyen 2004] that only MSC-1 can be used is incorrect. The findings of this thesis do not support Nguyen's claims either.

4.2.3. Packet loss rate

In the tests, all networks occasionally were able to yield error free transmission. Table 12 shows the average packet loss rates.

Table 12: Packet loss rates

Packet loss rate	**WCDMA**	**EGPRS 2+1**	**GPRS 3+1**
Average (%)	0.1	0.3	0.7
Minimum (%)	0.0	0.0	0.0
Maximum (%)	0.6	0.8	2.3
95^{th}-percentile (%)	0.2	0.3	2.2
Test runs (n)	25	32	32
Error-free runs (%)	55.0	56.7	30.8

The results show that more advanced PS networks are able to yield better quality, and WCD-MA is very close to being error free (when the signal quality is good). [Diaz 2007] tested streaming in a commercial WCDMA network in Spain and the packet loss rate was 0.58% on average. The results show that roughly 2 percent of the packets can be lost (Max GPRS loss rate). Such loss can significantly lower the service experience, but the losses are so sporadic that preventing the loss in advance may lead to significant overhead. RTP retransmission introduced in PSS Release 6 is a good solution to this problem.

The number of error-free test runs shows that almost 1/3 of GPRS tests were error free, while more than half of the EGPRS and WCDMA test runs were error free. The reason why the WCDMA percentage is lower than the EGPRS is due to the number of test runs in which only one or two packets were lost. If these test runs would have been error free, the WCDMA percentage would be 85%.

Weak signal quality can cause additional packet losses. Since cell reselections are triggered by weak signal quality, the error rates close to cell reselection were studied. Figure 11 shows the situation in GPRS with Reliability class 4. In period A (5 seconds before cell reselection) there is a clear increasing trend in packet losses and in the final second before cell reselection, on average, two packets were lost (10.7 packets were transmitted per seconds, which yields 18.6% packet loss rate at the edge of the cell) . During the cell reselection (period B) all the packets were lost, but this was omitted from the picture. After the cell reselection the signal quality is better and there should be fewer errors and period C (5 seconds after the cell reselection) supports this. There are still some packets lost during the first second in the new cell (0.3 packets on average, 2.8% packet loss rate), but after this, the situation stabilizes to being error free.

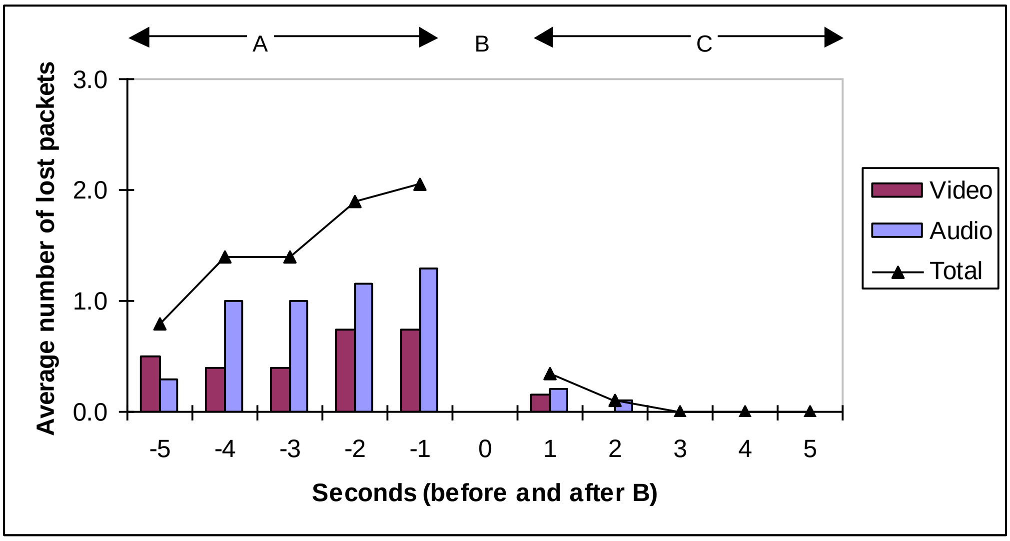


Figure 11: Packet losses around cell reselection

Cell reselection management proposed in Section 5.2.2. can handle the data losses during the cell reselection (Period B), but if the packet losses grow too high before cell reselection (period A) and there is no other means to recover the losses (e.g. RTP retransmission), a streaming client could request the cell reselection management to start the retransmission earlier (e.g. B-2 seconds). A server would not know that the client is concealing the cell reselection start time, but the client would also receive these packets that are lost during a weak signal period. Naturally, buffer levels needs to be taken into account, since this would lower the buffer levels.

4.2.4. Cell reselection time

In GPRS and EGPRS, the network chooses the time to change the cell. EGPRS may also include a Network Assisted Cell Change (NACC), but this was not used in the test. In WCDMA the decision of cell change occurs in the network and due to code-division, WCDMA is able to make a soft-handover where frequency does not change. The change of the frequency is the main cause of data flow break in cell reselection. Cell reselection time is a period when no data is received from the network. The typical delay between two consecutive RTP packets in our tests was on average less than 100ms, so it is easy to detect the cell reselection time even if no packet were lost (e.g. GPRS RC2). In [Gurtov 2002] GPRS, cell reselections were tested while driving in the city center of Helsinki. The interval of cell reselection was from 40 to 70 seconds, and the duration was typically below 5 seconds. Similar results have been achieved in [Rexhepi 2004], where streaming was tested in a live GPRS network while also driving around the downtown area of Helsinki. In their tests, the cell reselection varies between 1.8 and 3.3 seconds. GPRS test results (Table 13) show that the average cell reselection time increases when error protection improves. This is due to the fact that improved error protection requires more complex methods, which causes additional delays. Cell reselection occurs within one base station between two cells (Inter Cell CR).

Table 13: GPRS cell reselection times with different Reliability class

Packet flow stop	***RC 2***	***RC 3***	***RC 4***	***RC 5***
Average (s)	3.7	3.4	2.3	2.2
Minimum (s)	2.8	1.6	2.1	1.5
Maximum (s)	5.1	8.2	2.6	2.6
Standard dev. (s)	1.0	2.75	0.2	0.5
Test runs (n)	10	10	10	10

Table 14 compares the cell reselection times in different networks. The more complex nature of EGPRS (compared to GPRS) is visible in the cell reselection lengths. The target delay in EGPRS NACC is less than 2 seconds with a theoretical minimum of 0.7-1.2 seconds [Rexhepi

2005]. An interesting finding in the WCDMA tests was that the cell change period was not visible at all. Even though it was expected that no data would be lost during cell change, it is not possible to observe any variation in packet arrival delay, which would indicate when the cell change occurred.

Table 14: Cell reselection times

Cell reselection	**WCDMA***	**EGPRS 2+1**	**GPRS 3+1**
Average (s)	0.0	2.9	2.3
Minimum (s)	0.0	2.7	2.1
Maximum (s)	0.0	3.4	2.6
Standard dev (s)	0.0	0.2	0.2
Test runs (n)	10	10	10

*Only soft handovers

Inter-system handovers (ISHO in next table) between 2G and 3G are much more demanding than cell reselection within the network. The most common reason to ISHO is the mobility of the mobile device (i.e. the device either enters or leaves 3G coverage). Another reason for 3G to 2G ISHO could be the capacity limitation of a 3G cell, which causes lower priority services to be transferred to a 2G cell. Table 15 shows results of handovers made to both directions indicating that the handover from 3G to 2G is more demanding and takes more time.

Table 15: Inter-system handover times

ISHO	**2G->3G**	**3G->2G**
Average (s)	8.2	9.7
Minimum (s)	6.3	7.3
Maximum (s)	11.8	12.0
95^{th}-percentile (s)	11.2	11.8
Test runs (n)	10	10

[Bhebhe 2008], [Johnson 2005] and [Solhjoo 2003] have performed similar tests with slightly different delays. Bhebhe et al. had similar finding, reporting that 3G to 2G ISHO is more demanding although the delays were smaller (5.7 seconds ISHO from 2G to 3G and 6.8 seconds ISHO from 3G to 2G). In the test done by Johnson et al. the 3G->2G ISHO took 10.7 seconds which is consistent with the findings of this study. Test results from Solhjoo et al., however, indicate that 2G->3G handover takes more time (11.3 seconds) than 3G->2G (7.7 seconds). It is unclear whether these differences are caused by the network or different testing conditions, but it is clear that the ISHO delays are significantly higher than normal cell reselection and can cause problems in streaming quality.

Packet-switched handover specified in 3GPP [3GPP-9 43.129] is estimated to drop the delay near one second [Bhebhe 2008]. This new method allows a target cell to allocate needed resources before cell changes, in addition, more information about the target cell and mobile device is signaled before cell change.

4.2.5. Buffering requirement

There are two reasons for using buffering in streaming. The first reason is related to compensating the variation in packet arrival delays, and the second is related to hiding cell reselection. Cell reselection times were described in the previous section. Table 16 shows the packet arrival delay and variation.

Table 16: Packet arrival delays

Packet arrival	WCDMA	EGPRS 2+1	GPRS 3+1
Average (ms.)	82.1	83.7	84.9
95th-percentile (ms.)	186	434	671

The difference in average values is very small, which indicates that the majority of the packets arrive rapidly and with low jitter. But the 95th-percentile shows that there is a clear difference between these networks, and the GPRS causes much higher jitter values. The same can be seen when analyzing delays in the ISHO case (Figure 12). The delay variation (jitter) is much higher in 2G (GPRS) than in 3G (WCDMA)

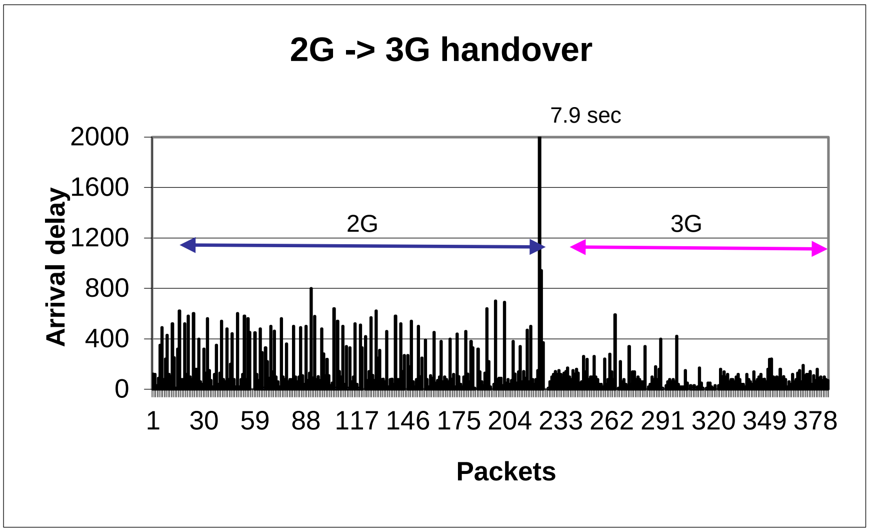


Figure 12: Packet arrival delay in ISHO case

In [Diaz 2007] the average delay of GPRS was 100 ms, with the 95th percentile at 120 ms. Unfortunately, they did not measure buffer sizes in their study. In [Hoymann 2002] the acceptable buffer for streaming is 10 seconds which allows retransmitting erroneous and lost packets. Table 17 shows how many seconds are required to buffer if only delay is taking into account and also if delay and cell reselection needs to be handled.

Table 17: Buffer sizes

Buffer length	**WCDMA***	**EGPRS 2+1**	**GPRS 3+1**
Delay only (s)	3	3	8
Delay and CR (s)	3	8	12

*Only soft handovers

Results show that the additional jitter caused by GPRS requires that a larger buffer is used and cell reselection requires 4-5 seconds additional buffer (in order to avoid rebuffering). If ISHO delay is included in buffering then the buffer length would be between 15-20 seconds.

The total service startup time is received when the RTSP signaling times and buffering requirements are summed together. E.g. in EGPRS, it would be around 15 seconds (7.3 s + 8 s). It is possible to lower the buffering time without lowering the buffer level. “Fast-forward buffering” described in Section 5.2.4 allows the server to send data faster than the client consumes it. In this case, the client can start the playback before the desired buffer level has been reached and the buffer level continues to increase during the first few seconds of the streaming. The drawback of this method is that the beginning of the streaming session needs to use lower media quality. The switch to normal media quality can be done after the buffer level is full.

In conclusion, the non-bitrate-guaranteed PS networks are challenging from a streaming point of view, but it is possible to maintain adequate service. Some studies (described earlier in this chapter) report that the transmission may be too erroneous for streaming, but they may overestimate the error level of the network, since commercial networks do not show such a high error level. It is clear that bitrate fluctuation and transmission gaps caused by cell reselection are the most critical problem in the streaming service, therefore, the next chapter provides solutions to these issues.

Chapter 5

Enhancements to PSS

IT was shown in Chapter 4 that streaming (with low quality) is possible with a basic PS network, but there are many areas in which it is possible to improve the experience. The simplest improvements change the existing network QoS parameters to be more suitable for PSS, but these improvements do not solve all the issues. Novel ideas are also required to improve the usability in areas of bandwidth adaptation and mobility management.

5.1. SESSION SETUP

The entire streaming session is defined in a setup and therefore it is important to understand what parameter values should be used. The GPRS specification [GSM 03.60 Rel 97] divides the traffic into two categories (simplification of Table 13 of the specification):

Real-time	Shorter delay	Must tolerate data losses
Non-real-time	Longer delay	Does not tolerate data losses

The problem is that streaming does not fit into these categories. It is real-time traffic, but it does not tolerate data losses. We propose a third category called Near-real-time that would characterize the streaming with moderate delay, but does not tolerate data losses. Therefore the proposal of the specification can be challenged, and the best parameters for streaming can be evaluated.

5.1.1. GPRS QoS parameters

In the GPRS network, the QoS parameter is limited to five major parameters. Two parameters describe the bandwidth (Mean throughput class and Peak throughput class). From a streaming point of view, these parameters are straightforward. They describe the content bitrate as it is and must be used as specified. The following examples can be used, when the parameters are set:

Values for low GPRS (2+1) bitrate would be:

- Mean throughput class 16 (~22 kbps)
- Peak throughput class 3 (32 kbps)

Typical GPRS (4+1) network requires:

- Mean throughput class 17 (~44 kbps)
- Peak throughput class 4 (64 kbps)

Theoretical maximum GPRS (8+1) bitrate requires:

- Mean throughput class 18 (~111 kbps)
- Peak throughput class 5 (128 kbps)

(Mean throughput class 18 is the highest this standard has, and Peak throughput class 6 (which would be the next value) is equal to 256 kbps, which is more than the theoretical bitrate of GPRS.)

Precedence and delay classes describe service commitments. Since streaming is a Near-real-time service, it requires that a network provides a best precedence and that the delays are moderate. A precedence class determines the importance of the service. If there are two services with different precedence classes competing for a limited network bandwidth, the service with the better precedence class gets the resource. The delay class indicates the maximum end-to-end transfer delay this service can tolerate. The following parameters should be used for streaming (since real-time services require best service commitment and shortest delay):

- Precedence class 1: Service commitment maintained before other classes.
- Delay class 1: Mean delay < 2 sec.

(Delay class 2 means the delay is < 15 seconds, which is too much for streaming)

The reliability class determines how well the GPRS network can correct errors before sending the data to the mobile device. This is the parameter that needs to be studied more to identify the best parameters for streaming. There are several values to choose from depending on the supported features. The specification [GSM 03.60 Rel 97] recommends that Reliability classes 4 or 5 are used for real-time traffic. These two reliability classes are not able to yield error free data transfer. The packet loss rate of the video sequence is around 1% [P1]. There is no reliability class (based on specification) that would be suitable for real-time traffic that requires error protections. Based on studies [P1, P2], the recommendation is to use Reliability class 3 or better (1 being most reliable). The studies revealed the following characteristics of the Reliability classes:

- Reliability class 3 allows maintaining error-free delivery during transmission, but service must handle cell-reselection data losses with other means.
- Reliability class 2 allows maintaining error-free delivery also during cell reselection (buffers the data during cell reselection).
- Reliability class 1 is not required, but it may improve media quality, if the network quality is low.

Due to the cell reselection, an 8-second buffer was used in the study, which is more than sufficient for handling the additional delays caused by improved error protection. Table 18 summarizes the findings of the required GPRS QoS parameters for PSS and shows the difference between specification value and the proposal based on test results (Table 12 on [P2]).

Table 18: GPRS QoS parameters for PSS

Parameter	Specification value	Test results
Delay class	1	1
Precedence class	1	1
Reliability class	4 or 5	2 or 3
Peak throughput class	4 or 5	4 or 5
Mean throughput class	17 or 18	17 or 18

5.1.2. EGPRS/UMTS QoS parameters

EGPRS provides much more parameters to choose from than GPRS. QoS parameters are specified for bitrate guaranteed networks and non-bitrate-guaranteed networks [3GPP 23.107 Rel 99]. Streaming is classified into a bitrate-guaranteed traffic class and naturally this should be used, if available, but due to the scope of the thesis, the non-bitrate-guaranteed parameters received more focus. An interactive traffic class was used in the test [P3, P4], since it provides the best QoS parameters for a non-bitrate-guaranteed environment. According to the specification, the class should be used for web browsing, and by nature it provides better error protections with additional delay.

Maximum bitrate parameter should be used according to the content (media bitrate + overhead). Absolute bitrate values can be used instead of throughput classes, which give more accuracy to the parameter.

With an interactive traffic class, it is possible to give traffic handling priority to the service, which affects the delays of the service. Since streaming is a near-real-time service, it is recommended to rate it as the highest priority. Allocation/Retention priority acts similar to a

Precedence class described in the previous section, and also with this parameter it is recommended to use the highest priority due to the real-time nature.

Error protection is divided into several parameters. Service Data Unit (SDU) error ratio indicates the number of detected errors, while the Residual error ratio indicates the number of undetected errors. The third parameter indicates whether erroneous SDUs are delivered or not. In the Streaming traffic class the SDU error ratio can be as high as 10^{-1}, while based on this study, it is recommended to use 10^{-4} (which is still possible also with a Streaming traffic class). If it is required that the network handles losses caused by cell reselection then 10^{-6} should be used, and this is not available in the Streaming traffic class anymore. The Residual bit error rate should also be near the best values that the specification offers (10^{-5} gives an error free transmission). Since errors propagate easily in video (due to prediction between frames) there is no reason to send erroneous SDUs. There are methods, like UDP-Lite, that would allow sending erroneous SDUs, but those are not part of PSS and therefore not included in this study. The Streaming class is able to apply good error protection, and based on our study it should be used even though it may add delay.

Since the testing was limited to non-bitrate-guaranteed network environments, UMTS parameters do not vary from EGPRS parameters and can also be used there.

Table 19 summarizes the findings of the required QoS parameters used in [P3]. The new QoS parameters introduced in EGPRS are clearly improved with those of GPRS. Not only do they provide the possibility of describing the session better, but they also match better with the test findings. Even though the tests showed that strict error protection should be used with streaming, the proposed values are still within the boundaries of Streaming traffic class.

Table 19: EGPRS/UMTS QoS parameters for PSS

Parameter	**Streaming class**	**Test results (Interactive class)**
Maximum bitrate	<2048 kbps	<2048 kbps
Delivery order	N/A	N/A
Maximum SDU size	1500	1500
Delivery or err. SDU	No	No
Residual error ratio	From $5x10^{-2}$ to 10^{-6}	10^{-5}
SDU error ratio	From 10^{-1} to 10^{-5}	10^{-4}
Transfer delay	250 ms or higher	N/A
Guaranteed bitrate	<2048 kbps	N/A
Traffic handling priority	N/A	1
Allocation/Retention priority	1	1

The selected QoS parameters are the foundation of good media transfer in all three networks, but they do not solve all the problems in this area. The next section describes the key problems of mobile streaming and how to solve them.

5.2. MEDIA TRANSFER

The theoretical network bandwidth and practical network bandwidth are often different. The theoretical bitrate is the maximum value that the network can offer in optimal conditions, with the best signal quality and without any load. Even though the network would be able to give the maximum bandwidth, it is possible that the device is unable to receive it. A practical bandwidth can be considered an average of available bandwidths in long time windows. It is not the worst-case scenario, nor the best. Practical bandwidth takes into account the device capabilities and also conditions that affect the network bandwidth (load, signal quality, movement etc). Theoretical bandwidth is almost always used in commercials and guides, but practical bandwidth is the value that the service needs to be aligned with. Since this thesis began before the first GPRS enabled mobile phone was marketed, the bandwidth values have changed a great deal during the study. It is evident that improved network bandwidth improves media quality, but there are other issues that impact the media transfer.

The non-bitrate-guaranteed nature of the networks has two impacts on media transfer: Data can be lost, and available bandwidth may change at any time.

5.2.1. Data losses

Data loss in the PS network means a loss of a data packet. Some data packets may never arrive to the destination either due to a corrupted address, overloaded network or cell reselection of mobile client. Some packets that do arrive to the destination may contain corrupted data and are therefore discarded.

For a streaming client, the data losses are either recurrent minor losses or momentary major losses. In recurrent losses one data packet is corrupted or lost without clear reason (i.e. it is not possible to understand or detect the cause of an error from the streaming client). Often the reason is a momentary reduction of signal quality. At the edges of network cell, the quality might be lower and causing this kind of data losses. If network parameters are correctly chosen according to the service needs, these errors are rare or can be hidden from the user. In our tests [P1, P3, P4] a typical loss was less than 0.5%, and with the recommended QoS parameters it was 0%. In momentary major losses, there are either several packets lost in a row or the packet loss rate suddenly increases dramatically. The two most common reasons for these are cell reselections and change in either the network Coding Scheme or the number of available Timeslots. Since a mobile device initiates the cell reselection, it detects when it occurs. Also the network informs the mobile device, if the network characteristics change. Therefore the mobile device is able to detect the reason for the two most common momentary major losses and able to adapt to the new situation.

5.2.1.1 Reporting a data loss

In [P5, Lundan 2002] we tested a method to report recurrent losses. An RTP protocol specification [RFC 3550] offers a control protocol called RTCP, which sends feedback messages of received media. Specification limits the bandwidth of RTCP packets and allows for using only 5% of the total bandwidth (2.5% can be used by server and 2.5% by client). In addition to this rule, specification requires that the maximum time between two RTCP packets should not exceed 5 seconds, which means that in some circumstances the reported error may be more than 5 seconds old (RTCP transfer interval + transmission delay). An event-driven RTCP feedback method (Originally introduced in [RFC 4585], improved with [RFC 5506]) tested in [P5, Lundan 2002] tries to solve the problem that feedback is received too late. Every time a client detects an event, it tries to send an RTCP packet, but due to the bandwidth limitation it may not always be possible if there are many events within a short time period. After an event-driven RTCP packet is sent, the client calculates when the next RTCP packet can be sent, in order not to exceed the given bandwidth. For example, if a 64-kbps connection is used, it means that 2.5% bandwidth is 1.6 kbps. The RTCP packet size is 88 bytes, which leads to 0.44-second RTCP interval:

1600 bps/(88 bytes x 8) = 2.27 packet/s => 1/2.27 = 0.44 seconds

In the test, the 60-second sequence included 580 RTP packets, which means 9.7 RTP packets/second. With a 1%-packet-loss rate, the errors would occur every 10.3 seconds (assuming errors occur in constant intervals). With a 3%-packet-loss rate, the errors would occur every 3.4 seconds and the 5%-packet-loss rate leads to 2.1 seconds. In theory, this means that the constant RTCP feedback mechanism would be able to handle this kind of PLR easily, and at the same time, it indicates that the Event-driven RTCP feedback method has no bandwidth problems (i.e. there is no need to discard RTCP packets due to bandwidth limitation). However, assuming a constant error interval is not realistic in the mobile environment.

Our tests in [P5] with a natural error pattern showed that the Event-driven RTCP feedback method worked well in a 1%-packet-loss rate, but with a 3%-packet-loss rate, it was no longer possible to report every event, since the bandwidth limitation caused RTCP packets to drop. Even though the number of discarded RTCP packets is low (1.4 %), it is an indication that the limit of benefits of Event-driven RTCP feedback is near. Every time an RTCP packet needs to be discarded, it means that a constant RTCP interval would lead to better reaction of the event. Over 20 RTCP packets (14.3 %) were discarded when PLR was 5%. Further studies in [Lundan 2002] revealed that 2% packet-loss rate is the limit to reporting every event.

These tests were performed before GPRS networks were available, and there was no knowledge as to what the typical PLR rates for streaming in these environments are. [P5, Lundan 2002] assumed much higher PLRs than [P1] revealed. Conclusions about the suitability of event-driven RTCP messages for streaming were done based on current knowledge of packet-switched networks, and later, tests revealed that packet loss rates are much lower than previously thought; therefore, an event-driven RTCP method is a very promising means of reporting errors and other events in a streaming service. However, it does not change the fact that the method has limits in regard to high error rates, and after 3%, the benefits begin to deteriorate.

5.2.1.2 Quality of Experience reports

The standard RTCP feedback mechanism has limitations. The specification defines only two mandatory fields of the problems: generic packet loss and delay values. RTCP Receiver Report includes also optional profile-specific extensions, where new information can be added. In addition to this there is an optional Application-Defined RTCP packet (APP packet), where an application can freely define new fields. Even though these extensions exist, PSS standard is not using them widely (only APP packet type is used when reporting buffer status starting from PSS Rel. 6). Even though an Event-driven RTCP error reporting, described in the previous section, is able to improve the reporting accuracy, it does not change the collected data.

A novel method described in [Lundan 2005] aims to solve this problem by providing much more information on what occurs in a streaming session. In the beginning of the session, the streaming server can request, with RTSP messages, the quality metrics it wants to receive from the client. A client can accept the request as is or propose a subset (e.g. if it does not support all the metrics). During the streaming session, the client can send the metric information via RTCP or RTSP messages. Depending on negotiated parameters, the values can be pre-calculated (e.g. average packet loss rate) or absolute values (number of lost packets). It is also possible to separate different media types (audio or video) and receive separate data. The method also provides an opportunity to renegotiate the metrics used, as well as stop metric sending during the session.

As an example, three SDP attributes were extracted from an RTSP message:

```
a=QoS-Metrics:{RTSPSetupTime, InitialBufferTime};rate=End
a=QoS-Metrics:{Framegab_Max, Framegab_Ave};rate=15;range:ntp=0-40
a=QoS-Metrics:{Audiogab_ave, Audiogap_max};rate=20
```

(The complete message can be found in [Lundan 2005])

The first attribute is presented in the common area of the SDP, and it describes the session level metrics. The rate value *End* indicates that these values are sent only once at the end of the session. The second attribute is presented in the video portion of the SDP message and it describes desired video metrics. The rate indicates that the metrics are sent in 15-second intervals and the range indicates that the metric sending starts from the beginning and is stopped after 40 seconds. The third attribute is presented in the audio portion of the SDP message indicating the rules of audio metrics. The shortest metric sending interval is 1 second. If the rate is equal to 0, it indicates that the metric sending is event-based.

Reporting can be done with an RTSP Options message or RTCP message and it contains one or more lines like (in RTCP format):

```
Audiogap_ave 105.5 6000
Audiogap_max 123 500
```

Where the first number is the metrics value and the second number is the timestamp. In the example, the report is sent with timestamp 6.000 ms with an average gap value of 105.5 ms, which indicates that the maximum gap of 123 ms occurred with timestamp 500ms. If the metric period contains more than one event, these can be listed (example in RTSP format):

```
Acorruption_dur={97 6000, 221 11000}
```

This reporting mechanism has several benefits over its predecessors (standard RTCP and Event-driven RTCP)

- It does not limit the collected parameters, but gives a mechanism to introduce new parameters.
- Due to the negotiation at the beginning of the session, it is possible to discard unknown parameters and adjust the sending frequency.
- It is possible to change the reporting in the middle of the session.
- There is no limit on how often a report can be sent (when RTSP is the protocol used)

3GPP Release 6 [3GPP-6 26.234] included this feature in the specification and refers to it with the term Quality of Experience (QoE) and with the following parameters:

- **Corruption duration:** Time of lost or corrupted frames
- **Rebuffering duration:** Time taken for rebuffering
- **Initial buffering duration:** Time of the first buffering
- **Successive loss of RTP packets:** Number of successive lost RTP packets
- **Frame rate deviation:** Difference between target frame-rate and actual frame-rate
- **Jitter duration:** Difference between target playback time and actual playback time

Some parameters are purely statistical such as the Initial buffering duration, but some could be used for reacting to real-time problems such as Successive loss of RTP packets.

5.2.1.3 Recovering from data loss

When data loss occurs, which in streaming cases means loss of an entire RTP packet, the amount of lost data is so high that recovery actions are required if the loss needs to be concealed from the end-user.

The most straightforward means would be to retransmit the lost packet, and the RTP has its own retransmission mechanism to handle these [RFC 4588]. One critical limitation of retransmissions is that there needs to be enough time to report the loss and receive the retransmitted packet. If the client buffer is small or buffer level is low, the playback time of the data may be over before the retransmitted data arrives. Retransmissions may also cause bandwidth overhead if packet losses occur frequently.

Forward Error Correction (FEC) [RFC 2733] includes the repair data into the media stream, so that the client can correct the error without requesting help from the sender. The benefit of the FEC is that recovery can begin immediately after the data loss has been identified. Since the packet loss rate is typically below 1%, this means that 99% of the packets would carry FEC data that serves no purpose, but the overhead is constant and easily calculated. According to [RFC 2354], FEC is more suitable to handle steady packet losses and retransmission should be used if data loss has a bursty nature.

The third method is interleaving [RFC 2354], which is not exactly an error correction method, but is targeted to minimize the effect of the loss. Interleaving means for example that the data is spread out in several packets or packet transmission order is changed not to be consecutive. Instead of including the entire video frame in the RTP packet, it would include a quarter of a frame from 4 frames. If the RTP packet is lost, then ¾ of the frame data would still be available (although there would be 4 frames without full data). Which of these two scenarios is better, is codec dependent. In audio coding the spreading can be time-based so that the RTP packets would not include a consecutive audio components, but there would be gaps (e.g. RTP1 includes moments 1, 3, 5 and RTP2 moments 2, 4, 6), which would allow a user to hear at least partial words.

Schorr et al. [Schorr 2004] compared the retransmission and FEC methods. Even though the tests are made in ISHO between WLAN and GPRS, it shows some differences between these methods. Comparing the video quality (by using PSNR), there was no clear difference between these two methods. If the buffer length was reduced from 3 seconds to 1 second, the retransmission method was unable to fix the losses. There was a significant difference in data rates. The FEC stream was 2.5 times larger than the original datastream while retransmission caused only a 1.5-times-larger stream (with a 0.2% error rate). It must be noted that the FEC implementation on [Schorr 2004] was very basic and FEC can be improved and overhead reduced. One such improvement is adaptive FEC overhead control mechanism studied by Kang and Koguinov [Kang 2005].

Combinations of different error correction and concealment methods have inspired many studies, and methods have improved significantly from basic algorithms. In [Korhonen 2008], the combination of packetization and FEC was studied. The data was split into small units called a Network Abstraction Layer Unit (NALU). This segregation into small units allows the RTP packets to be close to a constant size (i.e. the size variation between RTP packets was small), which reduces the overhead caused by FEC. Also the NALU units were interleaved into different packets to reduce the effect of packet loss. Each packet consisted of 3 NALU units and 2 FEC units (which are not related to the NALU units of the same packet). Small NALU size caused a larger header overhead, but this was mostly compensated by a smaller FEC overhead. Tests indicated that the small NALU units improved the video quality.

Another packetization method was investigated in [Superiori 2008a]. In this method, the information of the packet is grouped based on the information type and not based on the macroblock it belongs to. For example, all the characteristics of the macroblocks are grouped together and all motion vector data consist of another group. The bits indicating the basic characteristics of the macroblock are stored at the beginning of the packet, since it is statistically more protected than the end of the packet. For each group, it can be separately decided whether errors are accepted or not. For example, if there is a bit-error in the most critical groups, the packet is discarded, but for example a motion vector group can be partially used even with errors (those motion vectors that appear before error).

[Nemethova 2006] tested and compared several error concealment methods including different types of interpolation, boundary and block matching. The interesting part is not the methods applied alone, but the way in which they are applied together. A combination of different methods creates a decision-tree, where the best error concealment method is applied to certain data. For example if an error occurs in a scene cut, interpolation is used, but if the error is in an I-frame, then the method used is block matching. The tests showed that the decision-tree method improved the received video quality.

An improvement in retransmission methods was introduced in [Korhonen 2005], who proposed splitting the data to higher priority and lower priority packets and transferring the higher priority packets earlier than corresponding lower priority packets. This allows retransmission to have more time to correct data losses of higher priority packets. Even though the method is tested with audio streaming, it is easy to find the same benefits with video streaming. The I-frame and P-frame containing a scene cut could be a higher priority, while the rest of the P-frames (and B-frames) would have a lower priority. [Korhonen 2005] also studied the possibility of adding prioritization to redundancy-based transmissions. Data that is classified as a higher priority would be replicated to later packets that can be used in case the original packet is lost. The methods were tested with extreme packet loss rates where prioritized retransmission was a better solution than prioritized redundancy (the method was subjective listening test).

5.2.2. Cell reselection handling

Cell reselection occurs, when the mobile device notices that the neighboring cell provides better signal quality than a given current cell. During cell reselection, no data is sent from the network to the device, and based on selected QoS parameters, the network can either store the packets and send them after cell reselection or discard them. In a GPRS environment, the average cell change was 3.4 seconds [P1] and in EGPRS, it was 2.9 seconds [P3]. WCDMA networks can do soft handovers that do not cause data losses [P4].

A hard cell change (changes in cell frequency) causes a short period when no data is transferred from the network to the device. In the beginning of the session, a device can request a QoS parameter to handle this (e.g. GPRS Reliability class 2), but a novel solution in which a device and server can handle this without network parameters is presented in [Lundan 2004, P2]. In this method a streaming player detects that there is a time period when no data is received from the network. There are several methods for detecting cell change:

1. Since a device chooses a cell reselection, it could inform the streaming player that cell reselection is impending.
2. There is a certain average packet arrival delay (delay between two consecutive RTP packets). If the delay is more than the normal maximum delay (threshold can be adjusted), the device can trigger it as cell reselection.
3. With the correct QoS parameters, the RTP packet loss rate is very low. If there is a sudden increase in the packet loss rate, the device can trigger a cell reselection.

The first method would be the most accurate, but since it requires changes in the mobile devices, it is not the easiest to achieve. The second method uses a timer to detect whether a packet has arrived (this was used in [P2]). The third method is also able to detect other momentary major losses in addition to cell reselection changes. During cell reselection, the device does not receive any RTP packet, therefore, the second method uses this as a trigger. However, if there is some other reason for momentary major loss, which causes a high packet loss rate, but packet arrival delay may not be long enough to trigger the algorithm, then the third method could be used to request resending. The third method can only be used in momentary losses, since the quality must be normal when the resending is requested (there is no reason to request resending if the quality is poor during resending). If the PLR lasts a long time, then the bandwidth adaptation described in next section should be used.

After the client has detected the cell reselection, it requests the server to pause the data transfer (RTSP PAUSE request) and request to start resending (RTSP PLAY request) the data that was lost. Examples of such Pause and Play messages are:

```
PAUSE rtsp://example.com/foo RTSP/1.0
CSeq: 6
Session: 354832

PLAY rtsp://example.com/foo RTSP/1.0
CSeq: 7
Session: 354832
Range: npt=28.00-
```

The Range in Play message indicates the moment when the cell reselection started. Due to the RTSP specification, the stream must first be paused before resending is requested. Otherwise, the server would interpret the play command as a new session and not a continuation of the old session.

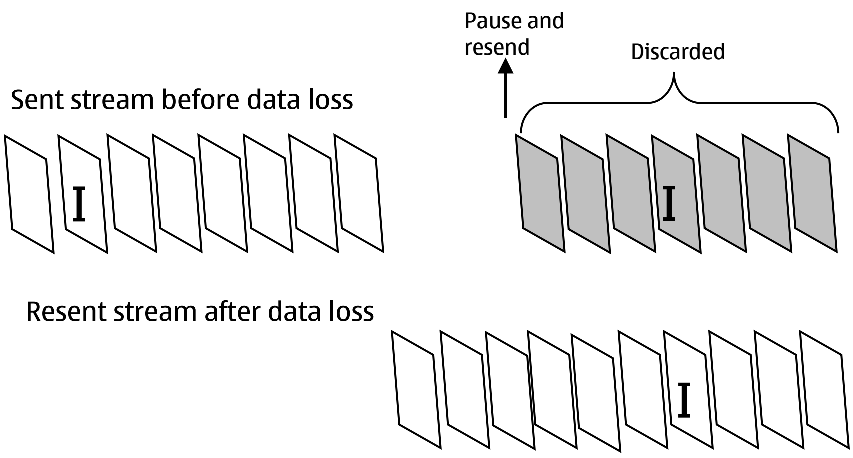


Figure 13: Cell reselection handling in streaming

Figure 13 illustrates cell reselection management at the video frame level. Even though a client sends a pause message to the server, it does not mean that the client must pause the playback. Playback can continue as long as there is data on the buffer. The play message can be sent immediately after the pause message. Data received after the pause message is sent, can be discarded since the data is resent anyway. It may take a few seconds to deliver the message to the server (see Section 4.2.1), thus it is not possible to know in advance how many data packets need to be discarded. The client can identify the last received RTP packet before cell reselection as well as the first resent RTP packet (i.e. the time set in the Range parameter).

The client can delete every RTP packet received between these two in the buffer. If the client buffer has more data than what is lost and discarded during the cell reselection, the end-user does not even notice that cell reselection occurred. Figure 14 shows the quality of the stream. The Peak Signal to Noise Ratio (PSNR) indicates the difference between a sent and received sequence. If the PSNR value is low, this indicates that the received sequence has a lower quality than the sent sequence. The two PSNR values (with and without cell reselection (CR) management) follow each other until the cell reselection occurs and the buffer drains around frame 400. The video quality begins to drop rapidly (black lines differ from the pink line). Since there are no more data on the buffer, the client shows a still image (last frame) and the difference between this frame and the correct frame grows larger (fast decreasing black curve). When the sequence changes, there is substantial drop in the quality. The final phase (the lowest period) is when the cell reselection and rebuffering already have been completed and streaming client displays data, but there are severe prediction errors before first Intra-frame. The lines meet again around frame 650, where the Intra-frame is shown. During this entire period the quality without cell reselection management was on average 17.8 dB lower, and the maximum difference between these two lines is 33.9 dB. This is a substantial difference since a 5-7 dB change becomes visible to the eye [Liu 2007].

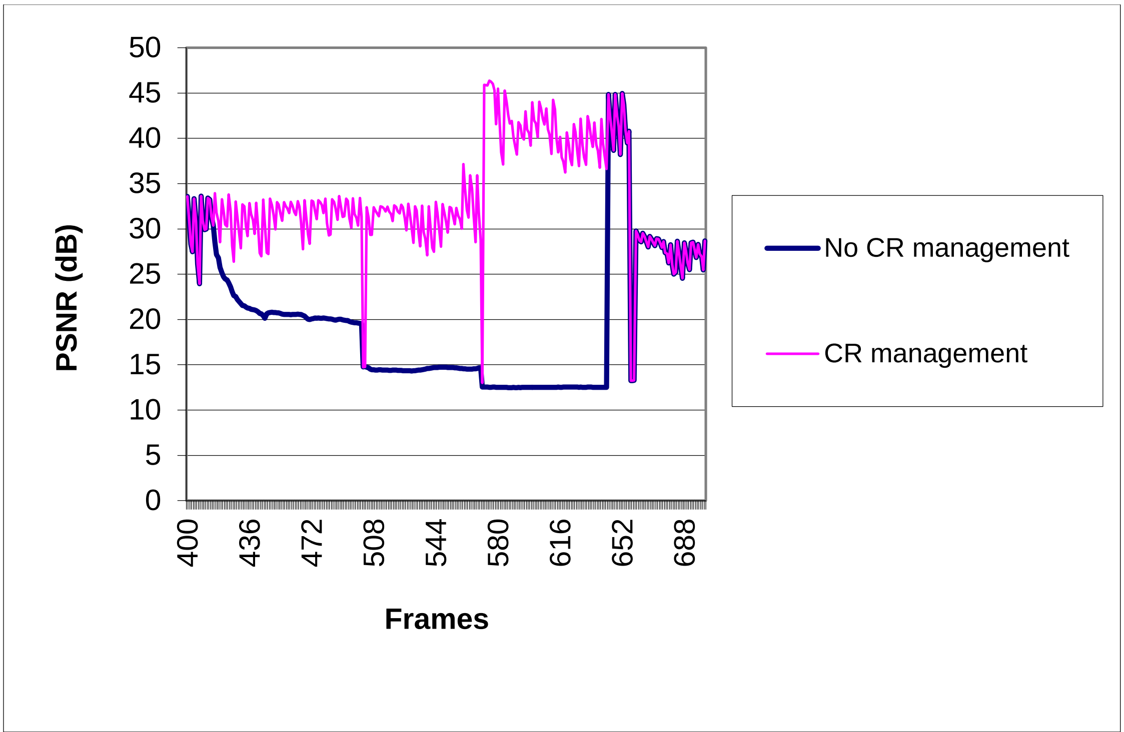


Figure 14: PSNR values in cell reselection

One can ask why not just request retransmission of the lost RTP packets alone? Why it is necessary to resend the entire stream and discard the previously sent data? The RTP Retransmission was included into PSS release 6, but there are certain disadvantages to this:

- RTP retransmission requires supporting new specifications, while the proposed solution relies fully on existing RTSP methods that must be implemented anyway, which makes the proposed solution simpler to implement.
- RTP retransmissions are done side by side with normal RTP transmission, which means that for a certain period of time, the streaming bitrate is two times larger, while the proposed solution maintains the original bitrate.
- RTP retransmission is only able to retransmit lost packets, while the proposed solution can be used to manage the buffer levels (See Section 5.2.4)

The 3GPP PSS includes only RTP retransmission (from Release 6 onwards) and no other means to handle cell reselection data losses.

Another reason to lose several video frames in a row is related to the network bandwidth drop. If the network bandwidth drops below the streaming bandwidth, the network starts to buffer the data packets that it is unable to send. Depending on the size of the buffer, the network can handle short-period problems, but after a while the network buffer overflows and data is lost. The solution to this problem is bandwidth adaptation which is described in the next section.

5.2.3. Bandwidth adaptation

The network cell has a limited amount of resources that need to be allocated to all devices. When the device leaves the cell, there are extra resources that can be given to other devices. When the new device arrives to the cell, existing devices may have to give some resources to the new devices (unless the resources are guaranteed to existing devices, which prevents the new device to get resources). Algorithms handling the resource reallocation on the network are often called rate adaptation algorithms and these algorithms that try to match the data transfer to the given resources are called bandwidth adaptation algorithms. Unfortunately, there is no consensus on the terminology, and the literature confuses these terms. This thesis concentrates on the latter part of the algorithms and calls these bandwidth adaptation algorithms.

In non-bitrate-guaranteed networks, the network bandwidth can change at any time. Network quality may require changes in the coding scheme and network load changes the available timeslots. Change can be positive (increased bandwidth due to more timeslots or decreased error protection) or negative (decreased bandwidth due to less timeslots or increased error protection). Negative network bandwidth change is easily detectable; if the bandwidth drops below the streaming bitrate, it starts causing data losses. Positive bandwidth change is not possible to detect by the end-users, but it means that they could receive better streaming quality than what currently is sent.

The media bitrate should change accordingly, when network bandwidth changes. It is not in the scope of this thesis to compare or improve methods to change media bitrate, but brief description of the most common methods is given. The easiest method to develop is to encode the sequence with different bitrates and then the server changes the whole bitstream, when media bitrate needs to change. We used this method to verify our bandwidth adaption algorithm (see results from Table 23). Some codecs (like H.264 [ITU-T H264]) allow more sophisticated methods. Scalable video coding allows different methods to change the media bitrate without changing the bitstream. Temporal scalability allows dropping whole frames (i.e. change frame rate) to adjust the media bitrate. Spatial scalability allows changing the size of the frame to adjust the media bitrate and quality scalability allows changing the quality of the frame. It is also possible to use all of these methods in one bitstream.

The basic bandwidth adaptation algorithm detects the loss rate of the data and if the loss rate is above a certain threshold, it lowers the data transfer bitrate. If there are no losses detected, then the algorithm behaves as a trial-and-error method and increases the transfer bitrate until losses begin to occur. There are two obvious problems with this algorithm: 1) when the error is reported, it is also detected so the end-user notices the problem before the adaptation begins. 2) this trial-and-error method causes additional error when there are no resources available.

The next level of algorithm tries to foresee the problems by measuring the buffer levels [Chemiakina 2003, Nilsson 2003]. If the buffer level begins to decrease, it indicates that the data sending/receiving rate is higher than the data transfer rate and should be lowered. This only solves part of the problem since buffer levels are steady if the transfer rate is higher than the sending rate and again a try-and-error methods need to be used to increase the bitrate.

Another approach for measuring the transfer rate is to calculate the delays from server to client, as in the TCP Friendly Rate Control (TFRC) [RFC 3448]. In [Alexiou 2006], TFRC was tested to estimate the network resources, and it was able to detect transfer rate increase and decrease.

Also, the network can provide additional information for bandwidth adaptation. [Antoniou 2007] proposes a method by which network congestion is used to detect the problem before it reaches the streaming client.

The bandwidth adaptation method described in [P6] is network initiated and implemented in the Radio Network Controller (RNC) of mobile networks. If the other end is in a fixed network then the algorithm is in its gateway. An algorithm calculates a data loss rate in two time windows, where the recent time windows is assigned greater weight than the older one. The purpose of weighting is to react faster to the changes while two time-windows try to reduce the number of unnecessary adaptations (bandwidth change can be very short, and the algorithm should not react to change unless it lasts long enough). Also hysteresis can be used to reduce unnecessary bitrate changes. Since it is not possible to detect bandwidth improvements with the packet loss rate, the algorithm is opportunistic: if there are no data losses, the detected algorithm attempts to change to a higher bitrate until it reaches the maximum bitrate or data losses begin to occur. If data losses occur, it lowers the bandwidth until the data losses stop or the minimum bandwidth is reached. Although [P6] uses AMR as an example, the algorithm itself is not audio specific and also works with other media types.

The aim of the first test was to demonstrate that the algorithm lowers the packet loss rate, if the bandwidth is reduced. Table 20 summarizes the results

Table 20: Network-based bandwidth adaption

Case	**PLR with 0.8 kbps lower bw**	**PLR with 4.8 kbps lower bw**
No enhancement	1.4%	4.4%
Enhancement	0.3%	1.4%

Since the algorithm is based on detecting packet losses, it is unable to yield loss-free data transfer, but the results clearly show that it improved the quality, since the packet loss rate is significantly lower.

The second test tried to find the correct size of the time window. If the time window is too short, then the algorithm is unable to detect the corrective actions and makes a second request before the first request is fulfilled. If the time window is too large, then the algorithm reacts too slowly and error remains noticeable too long. Table 21 summarizes the findings.

Table 21: Time window

Window size (T)	**0.8 kbps lower bw**		**4.8 kbps lower bw**	
	Avg bitrate	**PLR**	**Avg bitrate**	**PLR**
250 ms	11.6 kbps	0.3 %	10.1 kbps	1.4 %
500 ms	11.8 kbps	0.3 %	9.4 kbps	1.5 %
1000 ms	11.7 kbps	0.3 %	9.4 kbps	1.8 %
2000 ms	9.1 kbps	1.4 %	9.0 kbps	1.8 %

Four windows sizes (T) were chosen between 250 ms and 2000 ms. The algorithm uses a period of 2xT, which means that the overall time period was between 500 ms and 4000 ms. The algorithm used a weighting factor that gave the most recent time window 70% weight and the older 30% weight. The results show that a 2000-ms time window is clearly too large. A 1000–ms time window is already usable. The best time window depends on the error level. With a small error level (caused by 0.8 kbps lower bandwidth) the best time window is 500 ms and with a heavy error level (caused by 4.8 kbps narrower bandwidth) the best window is 250 ms.

All the mechanisms mentioned above are based on some estimation or calculation with thresholds and hysteresis done by the client or the server. They are all trying to figure out one value, the network bandwidth. But this information is already known by the network or the device. In [Chemiakina 2003] the network informs the server what the available bandwidth between device and base station is and the server can adjust the sending bitrate with an accurate network bitrate without any estimation errors or delays.

The novel method presented in this thesis is similar to the method described by Chemiakina, but the detection of the resource reallocation is performed by a mobile device instead of the network [P7]. The drawback with the network reporting the change is that the network must be taught to create RTSP messages, and the RTSP message must be made to appear as if it came from the device itself. A mobile device is able to create RTSP messages, and since the message is coming from an authentic streaming client, there is no need to change the identity of the message. A mobile device can detect the change, since it can determine what coding scheme is used and how many timeslots are available.

There are methods where a client requests certain bitrates from the server (for example the proposal of Ericsson Inc. in [Ericsson 020407]), which means this client must know what bitrates the server can offer, and the adaptation is simple but limited to certain bitrates. In a novel method, the client sends network bandwidth information to the server and then the server can decide what changes it will make to the stream. If the bandwidth has decreased only slightly, it might be enough just to skip certain frames, but if the change is more drastic, the server may change to a lower bitrate stream. The biggest difference between the methods described by Ericsson and the method described in this thesis is that in the Ericsson method the server does not know what the true network bandwidth is and whether that bandwidth is guaranteed or not. If the set of bitrates is predefined, then the client must always request the media bitrate that is below network bitrate. Table 22 provides an example of EGPRS network bitrates.

Table 22: EGPRS bitrate example

Network bitrate	**1 TS**	**2 TS**	**3 TS**
MCS-1 (kbps)	8.80	17.6	26.4
MCS-2 (kbps)	11.2	22.4	33.6
MCS-3 (kbps)	14.8	29.6	44.4
MCS-4 (kbps)	17.6	35.2	52.8

Assume predefined bitrates of 20, 30, 40, 50 kbps including all the overhead. The streaming starts with 3 TS and MCS-4 (52.8 kbps) and the server sends with a bitrate of 50 kbps. If more error protection is needed, the MCS is changed to MCS-3 (44.4 kbps). The method described by Ericsson requests the server to send at the bitrate of 40 kbps, while the method described in this thesis sends the information of the network bandwidth (44.4 kbps). In this case, both methods most probably end up sending with the same bitrate (40 kbps). Next, some resources need to be reallocated and there is only 2TS available (29.6 kbps). Ericsson's method sends 20 kbps of information, while the novel method sends 29.6 kbps of information. The difference between the network bitrate (29.6 kbps) and closest sending bitrate (30 kbps) is so small that the server should use 30 kbps instead of the next lower 20 kbps. The novel method does not consider all the possibilities for changing bitrates, it just offers a means to inform what the end result (i.e. bitrate) should be.

3GPP PSS Release 6 included a bandwidth adaption into the specification [3GPP 26.234 rel 6]. The difference between the novel method and specification method is in the means to detect the adaptation point. While the novel method relies on detecting the bandwidth change, the specification method relies on detecting changes in the buffer level. It should be pointed out here that the core idea of the novel method is to maintain the best possible buffer level and prevent buffer underflows and overflows. The specified method (in 3GPP) is relatively simple to implement and the buffer level monitoring can easily detect a negative bandwidth change, but the ability to detect positive bandwidth changes is questionable. Positive bandwidth changes (i.e. network bitrate is higher than streaming bitrate) are not visible at the buffer level, so the method either misses the opportunity to change to the higher bitrate or uses some try-and-error type of opportunistic method to guess whether bitrate is better than originally negotiated. This method was tested in [Huang 2008] and the test results clearly showed the drawbacks. When the network bandwidth dropped, the method reacted quickly, but since it does not know the true network bandwidth (only the consequences towards the buffer) the adaptation was not sufficient, but it had to perform a second adaptation before the correct bitrate level was found. Also, when the network bandwidth increased there was significant delay (minutes) in detecting this change. Unfortunately, the report did not clearly indicate how this improved bandwidth was detected.

Another study from [Curcio 2005] shows how this method reacts in cell reselection. The report shows that the method is able to react to cell reselection with certain limits. Again since the method does not exactly detect the cell reselection (but only the consequences for the buffer), the reaction depends on the buffer level. In the test, there were 3 cell reselections during the session. An algorithm detected the first cell reselection, but the reaction was very short, and it is unclear whether this reaction was necessary. The second cell reselection was not detected at all, but the third cell reselection was accurately detected and the length of the reaction matched the length of the cell reselection impact. In this report, the reaction times are much faster than in the previous report [Huang 2008]. Unfortunately the implementation differences between these studies have not been clearly defined to explain this situation. Based on these studies, it is clear that the 3GPP method can react to bandwidth changes, but the accuracy may need to be improved (or at least provide clearer guidelines and instructions on how to implement it).

In the novel method, the information of multiple sending rates can be sent with an RTSP DESCRIBE message by using the SDP protocol. As an example, there is an extraction from SDP:

```
a=fmtp:98 profile=0; level=10; bw=20, 30,40,50
```
or
```
a=fmtp:98 profile=0; level=10; multibw
```

Where the first line explicitly describes the bitrates used and second indicates that multibandwidth support exists.

The bandwidth change request can be sent with RTSP OPTIONS or SET_PARAMETER messages using the Bandwidth parameter already specified by RTSP.

The algorithm that sends the bandwidth change request is

```
If (Current_Bandwidth ≠Previous_Bandwidth) {
  Start timer Bandwidth_change_timer;
    If (Bandwidth_change_timer > k seconds)
      Send OPTIONS or SET_PARAMETER message with Bandwidth field.
}
```

The algorithm stores the bandwidth information and compares the previous bandwidth to the current. In order to avoid too frequent changes, a *Bandwidth_change_timer* is used. If there are no changes in the bandwidth in *k* seconds, then the RTSP message is sent with *Current_Bandwidth* value in *Bandwidth* field. If the new bandwidth change occurs before time *k*, then the timer is reset. An example of such a message:

```
OPTIONS rtsp://example.com/foo RTSP/1.0
CSeq: 7
Session: 354832
Content-type: text/parameter
Bandwidth: 27000
```

If the multirate bandwidth scenario includes different codecs, then the messages used needs to be different. The reason to use different codecs is that quality might be better with certain codec- bitrate pairs, or there is a simply a bitrate limitation that prevents using the codec. If more than one codec is used, the codec needs to be defined with RTSP DESCRIBE messages. Also according to the RTSP specification, the codec can be changed only with the RTSP PAUSE and PLAY messages. More examples of these messages can be found in [P7].

One additional element of bandwidth adaptation described in [P7] is that the network may change from bitrate-guaranteed to non-bitrate-guaranteed and vice versa. A client should send the information of changed QoS status to the server to improve the server capability to decide the proper sending bitrate. The message can be sent with bandwidth change message or independently if the bandwidth does not change. The bandwidth message together with the bitrate guaranteed parameter indicates to the server that the network bitrate was targeted to be steady (as long as the client stays in a bitrate-guaranteed network) and if there is need to change, the method can be such that it lasts longer. If there is no bitrate-guaranteed parameter in the bandwidth change request, the new change request can come at any time and adaptation method should not be heavy and time consuming.

In the tests, the Peak-Signal-to-Noise ratio was used to verify the algorithm. It allows revealing errors in video quality. Table 23 summarizes the results. Case 1 is the control test without bandwidth drop and adaption, which yield to PSNR value 30.8 dB. In case 2, the network bandwidth was reduced from 26.2 kbps (GPRS 2TS CS2) to 18.1 kbps (GPRS 2TS CS1) in the middle of the sequence. The end result was that 5.7 % of the video packets were lost, and the PSNR value dropped 6.6 dB (to 24.2 dB). Case 3 is the same as Case 2 from the network point of view, but the bandwidth adaption was enabled in the streaming client. Adaptation reduced the media bitrate from 20 kbps to 16 kbps after it detected the bandwidth change. Some packets were also lost in this case (2 packets/0.9%). Even though the media bitrate dropped 4 kbps, the PSNR change is not that dramatic (only 0.3 dB)

Table 23: Application-based bandwidth adaptation

	Media bitrate	**PSNR**	**Packet loss rate**
Case 1: No bandwidth reduction	20 kbps	30.8 dB	0%
Case 2: Bandwidth reduction without adaption	20 kbps	24.2 dB	5.7%
Case 3: Bandwidth reduction with adaptation	first 20 kbps, adapted to 16 kbps	30.5 dB	0.9%

The difference between these two bitrates (20 kbps and 16 kbps) is small, which explains why the PSNR figures are so close to each other, but this is consistent with the bandwidth drop in the network. On average, 5.7% PLR yields a 1.1-kbps media bitrate drop. Even such a small change was clearly visible in the video quality. The test sequence was the same 60 seconds H.263 encoded movie trailer described in Chapter 4.

5.2.4. Buffer filling

The cell reselection management described in Section 5.2.2 is able to hide cell reselection from the end-user, but the buffer is no longer full. Also bandwidth adaption described in Section 5.2.3 may not keep the buffer full although some methods monitor the buffer levels. An innovation described in [Lundan 2004] combines cell reselection management and bandwidth adaption to refill the buffer. The idea is to lower the playback bitrate below the transfer bitrate, but send the media still using the transfer bitrate, which kind of fast-forwards the buffer full. Figure 15 illustrates the issue.

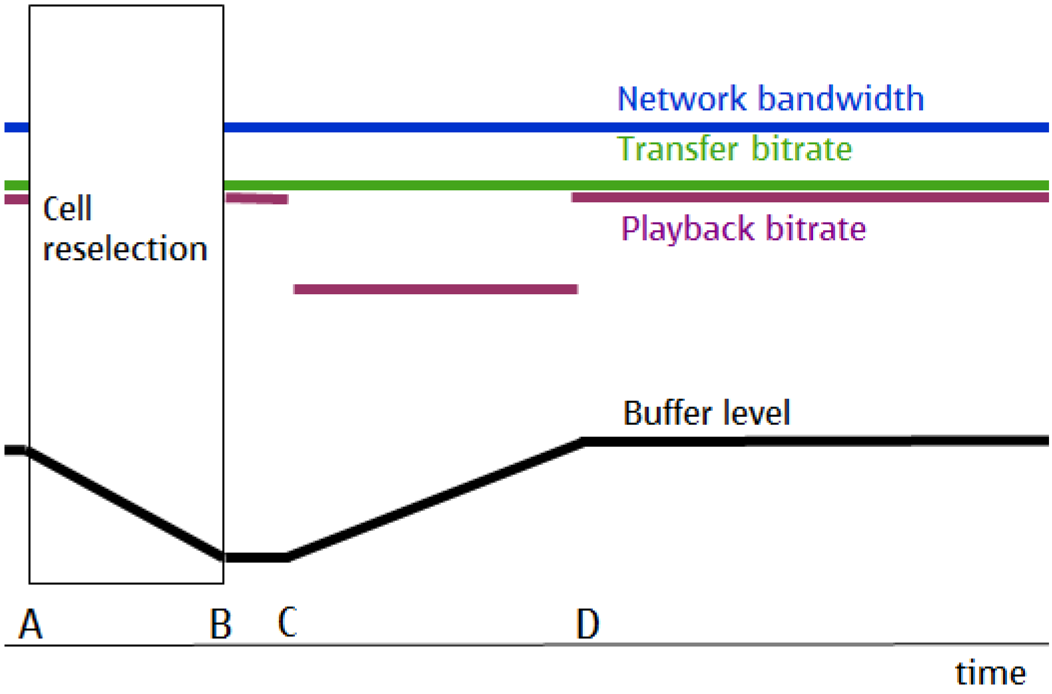


Figure 15: Buffer filling

Example: The server has two streams with playback bitrates X and Y, where X = 2xY. The streaming server sends the stream with bitrate X, which is slightly less than the available network bandwidth. The cell reselection begins at the time A, and the client notices that the buffer begins to drain. At time B, the client notices that the data starts to flow again and starts the refilling activity. It asks the server to change the stream from X to Y, but asks the server to send it two times faster than the playback bitrate (the transfer bitrate remains the same during the entire process). At time C, the buffer starts to fill. Therefore, every second that the client uses for playback, the server sends two seconds of data. When the buffer is back to an acceptable level, the client asks the server to change back to stream X and a normal sending speed. Since it takes little time to adjust the bitrates, the client must signal the change before time D, so as to not overflow the buffers.

Since the quality of Y is naturally lower than X, the end-user will receive lower quality data for a certain amount of time, but it is still less inconvenient than the buffer underflow situation where in the frame flow is halted completely during rebuffering, which may take several seconds.

The method can use the existing RTSP PAUSE and PLAY messages and Range, Bandwidth and Speed parameter in these messages. Example of the first message pair would be:

```
PAUSE rtsp://example.com/foo RTSP/1.0
CSeq: 6
Session: 354832

PLAY rtsp://example.com/foo RTSP/1.0
CSeq: 7
Session: 354832
Range: npt=28.00-
Bandwidth: 20000
Speed: 1.5
```

Where *Speed* 1.5 indicates that the *Bandwidth* 20000 (20 kbps) should be sent as 30 kbps. If the end point of the buffer filling is not known by this time, the parameter *Range* can be open (28.00-), but if the end point is known, it can be closed (e.g. 28.00-40.00).

When the buffer is near full, another message pair is sent to normalize the situation

```
PAUSE rtsp://example.com/foo RTSP/1.0
CSeq: 8
Session: 354832

PLAY rtsp://example.com/foo RTSP/1.0
CSeq: 9
Session: 354832
Range: npt=40.00-
Bandwidth: 30000
Speed: 1.0
```

The algorithm to determine the period can be:

$$filling\ period = \frac{LowSeqTime}{speeding\ factor}$$

wherein

$$LowSeqTime = (BufferSize - BufferData) \cdot \frac{original\ transmission\ bit\ rate}{original\ transmission\ bit\ rate - new\ transmission\ bit\ rate}$$

$$speeding\ factor = \frac{original\ transmission\ bit\ rate}{new\ transmission\ bit\ rate}$$

Where *BufferSize* is the size of the buffer in seconds and *BufferData* is the data left in the buffer (in seconds). *LowSeqTime* is the length of time the lower bitrate is visible to the end-user. As an example: a 12-seconds buffer that has 4 seconds left and an original bitrate of 30 kbps and a new bitrate of 20 kbps. The *Speeding factor* is 1.5 and *LowSeqTime* is 24 seconds, which makes the *filling period* 16 seconds.

Due to the predictive compression technique, there are P-frames that do not contain the entire data of the frame, but only the difference compared to the previous frame. If the P-frames do not match, the change between streams may cause prediction errors (i.e. the motion from the last frame in sequence X may not match the motion of the first frame of sequence Y). If the frame rate and bitrate difference between X and Y is not large, this is not noticeable to the end-user, but it must be noted that prediction errors have a cumulative nature (i.e. the error is visible until the next I-frame). To avoid the prediction errors, the change needs to be made with P-frames that match each other, or to the I-frame.

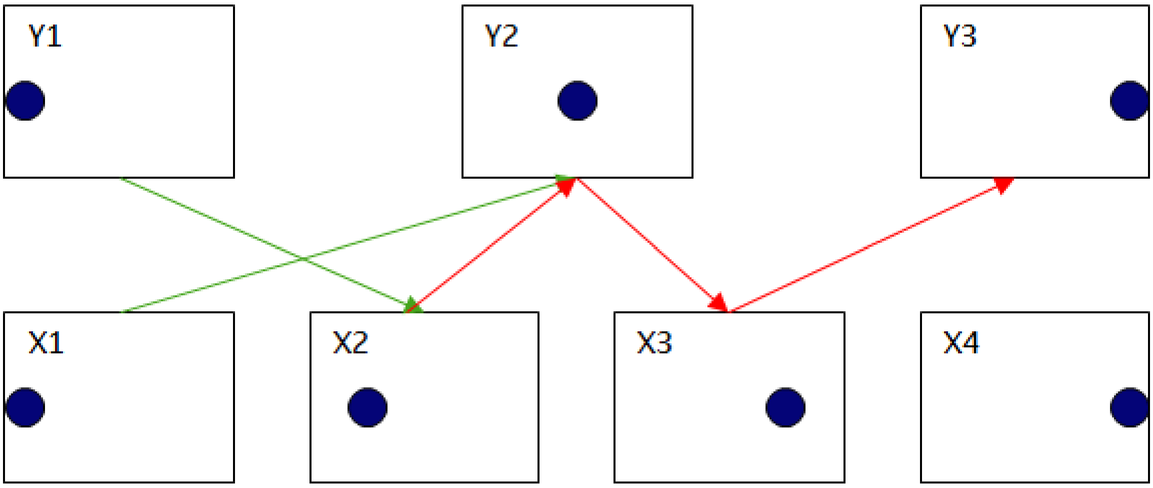


Figure 16: Sequence change

Figure 16 explains this scenario. Let us assume that all these frames are P-frames. The change from Y1 to X2 does not cause any prediction error, since Y1 and X1 match each other. The same applies to the change from X1 to Y2. The change from X2 to Y2 causes a prediction error, since Y2 contains difference data between Y1 and Y2, but the frame shown previously is X2, so the decoder draws the difference data from Y2 on top of X2. Same applies to the changes from Y2 to X3 and from X3 to Y3. In contrast assuming that Y2 is an I-frame, the situation changes. Now this change from X2 to Y2 (towards the I-frame) causes no error. However, the change from Y2 to X3 (away from I-frame) causes the same error as in the earlier case.

A 5-second Intra-frame rate was the most common in the tests. If it takes for example 7 seconds to fill the buffer, then the algorithm must make a decision:

1. Make the change in the previous I-frame and lose 2 seconds of filling time.
2. Make the change in the next I-frame and risk buffer overflow.
3. Try to find the best matching P-frames and apply the change to them.

Solution 1 is the simplest and safest, while solution 3 provides the best buffer level, but requires a more complex algorithm.

The algorithm of Solution 1 is:

$$SeqChangeTime = IFrameTime_{Original} * \left\lfloor \frac{SwitchTime + filling\ \ period}{IFrameTime_{Original}} \right\rfloor$$

Where *IFrameTime* is the interval of I-Frames, *Switch time* is when buffer filling begins and the *Filling period* is the time to fill the buffer (see algorithm earlier in this section). As an example, an I-Frame period of 5 seconds is common, thus let us use a previously calculated filling period of 16 seconds and assume that the buffer filling started at 28 seconds. The algorithm provides that the sequence should be changed back at the 8th I-frame with the time of 40 seconds. The algorithm indicates that buffer filling will last only 12 seconds instead of 16 seconds, which means that only ¾ of the buffer filling can be achieved. The buffer filling should have started at 30 second, which would allow a buffer filling period of 15 seconds, which is the best that can be achieved in these conditions.

5.2.5. Retransmissions on RLC layer

In an unpublished study by Miikka Lundan and Igor Curcio (from year 2003), the possibility of using lower network layer retransmissions to improve streaming quality was analyzed. The simulator described in [3GPP VCEG-M77 & 3GPP VCEG-N37] was used to simulate retransmissions in the Radio Link Control (RLC) layer. The simulator allowed defining the number of retransmissions. The study analyzed how well the retransmissions were able to hide packet losses and how much transmission overhead these retransmissions caused. In the test, we used an error pattern that caused around a 3% packet loss rate. This is a relatively high packet loss rate compared to our previous findings from real mobile networks, but we wanted to understand also the extreme situations and overhead in these cases.

Test results showed that the first retransmission was able to hide all errors caused by the error pattern. (I.e. none of the retransmitted RLC frames were lost). The simulated environment did not allow us to measure true delays caused by retransmissions, but hypothetically we can calculate an estimate. The average packet size in our tests was around 500 bytes and the RLC frame was 80 bytes, which means that data from one packet needs, on average, 7 RLC frames. If one of these RLC frames is retransmitted, it adds 14% extra transmission delay plus the delay caused to request the retransmission (since none of the retransmitted frames were lost, it must be very rare that more than one frame per packet is lost). Section 4.2.1 showed that it takes around one second to send an RTSP packet and receive a response, so if we assume that a sending packet from a server to a client takes 500 ms then a 14% additional delay is 70 ms. (RLC frame sending is not using the entire 500 ms, but for simplicity, let us ignore this fact). Our recommended buffer size is 3.000 ms, thus 70 ms additional delay can be easily hidden by the buffer. Even if a retransmission request would double the delay, it is still far from the buffer size.

Another measurement that was performed with the simulator involved determines the amount of bandwidth retransmissions require. We simulated this with a 64kbps bitrate, a 500-bytes packet size and 3% packet loss rate. Then allowing that one retransmission provided 1 kbps additional bitrate. This means that if we want to allow one retransmission, we would need to lower the original bitrate from 64 kbps to 63 kbps. This kind of change is not visible to the end-user and when it hides a 3%-packet loss rate that causes severe visible error, it is clear that retransmissions should be allowed.

5.3. SUMMARY

The streaming session begins by requesting Quality of Service (QoS) parameters that describe the characteristics of the session. We started this study by analyzing the proposed QoS parameters of the specification. The results show that the proposed specification values are otherwise correct, but in the Reliability class (which defines the error handling of the network), the specification proposal can be improved. According to the specification, Reliability classes 4 and 5 are meant for real-time traffic with low error handling capability. By using the specification values the tests of [P1] yield around 1% packet loss rate. According to the specification, improved error handling causes additional delay, which real-time traffic is unable to handle. Even though streaming is considered real-time traffic, the buffering mechanism allows for more relaxed delay bounds than other real-time traffic (e.g. video telephony), which allows for using improved error protection compared to what QoS specification proposes. With Reliability class 3, the streaming was error free and with Reliability class 2, the network buffered the data during cell reselection and made it loss free.

Compared to GPRS, in the EGPRS environment, there are other QoS parameters (that are common with WCDMA). This new QoS specification divides the traffic into four classes (Conversational, Streaming, Interactive and Background). A streaming service is meant to be used with a Streaming class, but based on our study, it is possible to stream with a less demanding QoS class. Again the difference is in error handling. The Streaming class allows requesting for less demanding error handling (Bit Error Rate (BER) $5x10^{-2}$ and Service Data Unit (SDU) error ratio 10^{-1} being the lowest). In our test, we used an Interactive class, which provides better error handling and noticed that it is possible to stream with more demanding error handling (BER 10^{-5} and SDU 10^{-4}). The same values can be requested also from the Streaming class, but they are on the upper limit of error handling: it is not possible to receive a GPRS Reliability class 2 type of error protection that hides cell reselections with the Streaming class. So based on our study, it is not mandatory to use a Streaming class to use a streaming service and client should request the best (or near best) error handling this class can provide.

Due to the strict delay bounds, retransmissions are typically not considered an option to improve streaming quality. Earlier studies already proved that it is not necessary to maintain such strict delay bounds and therefore retransmissions are a potential method. Retransmissions at the Radio Link Control (RLC) Layer were analyzed, and the study showed that first retransmission covered the losses caused by normal error level (i.e. error levels witnessed in our own tests). So there is no need to set the network to full RLC retransmission (i.e. retransmit packets again until it is received), because it is so rare that a retransmitted packet is lost. Allowing that one RLC retransmission does not significantly increase delay.

Lossy cell reselections of GPRS and EGPRS (and inter-system handovers in WCDMA) cause data gaps in streams. There are solutions to overcome this problem, including using Reliability class 2 or equivalent error protection, but there is also a novel solution that does not request anything from the network. In this solution, a client detects a gap in data flow. The method to detect the gap can be limited to cell reselections or made more generic to detect the gap without knowing the specific reason that caused the gap. When streaming client detects a gap, it waits until the data flow gets back to normal and sends two messages to the server to indicate what the last received data was and requests the server to start resending the data from that point. If the data flow does not normalize within a selected time limit (i.e. the problem is not a data gap, but continuous error situation), bandwidth fluctuation handling methods should be used.

Since this gap also drains the client buffer, there is a novel method to refill the buffers. Bandwidth fluctuation handling, described in the next section, requires that there is a possibility to change the content bandwidth. The buffer refill method requests the server to change from the current content quality to a lower content quality, but keep the original data-sending bitrate (i.e. kind of fast-forwarding the buffer full). When the buffer is full, the client requests the server to change transmission back to the original content quality. The drawback is that there is a moment of lower quality content, but this is far less of a problem than what would have occurred if the buffer underflowed. The switch between two different content qualities can occur at any time, but in order to reach a totally error-free switch, it should occur on an Intra-frame (a video frame that has no prediction data). The moment when the buffer is full may not always match with the time when the Intra-frame occurs, which leads to a dilemma: *should the client fill the buffer totally and risk reaching the sub-optimal switch point or request the* *server to switch content quality before the buffer is totally full in order to reach the best switch point?*

Basic packet-switched networks do not guarantee the available network bandwidth, but it can change at any time. Due to this fact, the media streams should adapt according to the available network bandwidth. One existing solution is that the server describes to the client what kind of content it has (same content with different bitrates) and if the bandwidth changes, the client can request the content bitrate that is most suitable. The novel method improves this solution by reporting the actual network bandwidth and allowing the server to decide what actions to take. Instead of blindly asking a certain bitrate and server to fulfill the exact request, the method simply reports the upper boundary that cannot be exceeded, and the server can then use several methods to fulfill the request more accurately. If the change of the bitrate is large, the server can switch to another content and even change the codec used. For smaller changes, it is possible to change only the audio bitrate or drop some video frames. This method also indicates whether the new bitrate is guaranteed or not, which may influence the adaptation mechanism used. This method is not limited to certain adaptive mechanism, but allows for using many.

Another novel method uses network monitoring to adapt the content. Network monitoring involves a certain time window of packets and detects packet losses. The time window is divided into two sections to give recent events more weight than older events. When the algorithm detects a certain amount of packet loss, it requests to change to a lower bitrate. It does not know what bitrate this is, but it just attempts to use the lower bitrate than the current. If the method does not detect packet losses, it becomes opportunistic and requests the next higher bitrate until it reaches a point at which errors start to occur. This method is most suitable with content that has many bitrates that are close to each other.

The RTCP protocol provides a method to give feedback to the server about transmission quality. In order to not to consume too much bandwidth, there is a limitation that feedback messages can only use 5% of the total bandwidth (2.5% uplink and 2.5% downlink). But a message should be sent at least every 5 seconds. Specification determines these rules, which results in lead to feedback messages with constant intervals. Existing improvement to the standard is called an Event-driven RTCP mechanism, which breaks the constant interval and sends feedback when certain event occurs. Since it still must obey the 5% rule, there may be situations in which it is not allowed to report an event. The purpose of this study was to determine the limit in which an Event-driven RTCP mechanism could be used. A 64-kbps sequence was used, which leads to maximum 3.2 kbps RTCP bandwidth and a 0.43-second constant interval. With the same bandwidth, the Event-driven feedback mechanism is able to report every event if the packet loss rate is 2% or under, but if it is over 2%, then some Event-driven RTCP messages have to be discarded. Even though Event-driven RTCP improves the accuracy of the feedback, it is still limited to report the data as defined in the RTCP specification.

In order to obtain much more comprehensive feedback, a novel solution was required that allowed streaming clients to create a better feedback message. In addition to the RTCP messages, RTSP messages could also be used. There are several benefits to using RTSP in feedback. With RTSP, it is possible to negotiate what feedback is gathered in the session setup. The streaming server can request certain feedback information from the client and the client can describe whether it can fulfill the server request or not. If RTSP messages are used as feedback messages, there is no limitation on how many messages can be sent (as is the case with RTCP). The method includes also many new parameters that the traditional RTCP feedback method is unable to describe.

Chapter 6

Media perception in mobile devices

HOW do people perceive media through a mobile device? There are many differences between traditional television viewing and consuming media through a mobile device. The following list summarizes the obvious differences:

Large display size vs. small display size

Long viewing distance vs. short viewing distance

Private viewing vs. public viewing

Low interruption level vs. high interruption level

Immobility vs. mobility

When viewing the list, one can easily see this issue is not as black and white as this list implies. The height of the average home-TV screen is easily 20 times larger than the height of a mobile device screen, which means that it is difficult to notice small details on such a small screen. On the other hand, the viewing distance is much shorter on a mobile device (typically at arm's length), which increases the relative size of the screen. If measured relatively, TV screen is just 2-3 times greater than a mobile screen.

Another aspect of this is the environment in which media is consumed. TV is usually watched at home, which is more or less private, and the typically level of interruption is low. If you are watching a video with your mobile device, you are probably on the go (e.g. sitting on a bus, waiting for someone in a shopping mall, etc.). There might be strangers sitting next to you and the environment could be full of events that easily interrupt the viewing.

In this chapter, we focus on the physical differences and leave the environmental issues for future studies. Two areas that we studied are lip synchronization and video quality in low-resolution devices.

6.1. LIP SYNCHRONIZATION

Lip synchronization involves simultaneous playback of audio and video. Another term to describe this is Inter-stream synchronization. People are quite perceptive in detecting incongruity in lip synchronization by simply watching people's lips. There are many methods to maintain lip synchronization, and Ishibashi et al. have done many studies comparing these methods [Ishibashi 2000 & Ishibashi 2001]. Since our aim is to find a subjective lip synchronization tolerance threshold for mobile devices, the technical comparison was not analyzed.

There are great deals of studies on this topic related to TV viewing. Most often cited is [Steinmetz 1996]. Based on Steinmetz, it makes a difference whether audio is delivered before or after the video. People are more tolerant if the video comes before audio, since this can happen in a normal everyday environment. If the speaker is far from the listener, their eyes can see the lip movement before the ears hear the sound. A 20-meter distance makes the visual arrive 60 ms earlier than the audio. There is no similar natural environment where the audio could arrive before the visual and people could detect this error easier than the visual before the audio error.

According to [Steinmetz 1996] people begin to recognize lip synchronization errors on TV if the error is ±80ms, and error becomes annoying after ±160 ms. Even though the final threshold shows no difference between audio early and audio late, more detailed data show that there is much more variation in audio-late test results. The test results also showed that the size of the head relative to the screen has some impact on the lip sync, but the impact is of the same magnitude as were individual differences among the test persons, and therefore, it did not change the threshold. Also the language used showed no difference. The test was done with three different sequences (Head, shoulder, and body) each 30-seconds long. Lip sync error was increased in 40-ms steps (±40ms, ±80 ms, ±120ms…) up to 320 ms. The sample size was 107 across all ages and sexes.

The Advanced Television System Committee (ATSC) has proposed much stricter limitations on lip synchronization. According to [ATSC IS-919] audio should never lead the video more than 15 ms and should never lag behind the video more than 45 ms. LinearAcoustics claim that video experts can detect 20 ms error in lip synchronization, and for example, the Dolby Digital synchronization target is a 5-ms audio lead and 15-ms audio lag [LA 2004]. The International Telecommunications Union (ITU) based their limitation on a subjective study and found that the detection point of lip sync error is a 45-ms audio lead and 125-ms audio lag [ITU-R 1359]

Our aim was to find a similar threshold for a mobile device; the details of the test setup and the results can be found in [P8].

The aim was to repeat the tests done by Steinmetz, but use a mobile device in viewing. The tests were simplified since only one sequence was used that resembled the Shoulder sequence of Steinmetz. The synchronization error increased in 40–ms steps, and only the detection threshold was reported. A total of 240 non-expert individuals participated in the test (20 persons per test set). Each test set included one sequence defined in Table 24. People were asked to report if they could detect some continuous error in the scene. The nature of the error was not revealed, so that people would not concentrate on lip motion more than in a normal viewing. The tests began with an error-free sequence, so that pure guessing could be eliminated. The testing environment was a normal office environment, and test subjects could freely choose the manner in which they preferred to watch the video. Typically they held the device in their hand, so the distance to the screen was 50-80 cm. We stopped the test when people found that lip synchronization was out-of-sync, therefore, in our tests, we found the threshold point at which the error was detected. We could have continued and asked whether the error was annoying and making the sequence worse, but we were only interested in finding the detection point.

Table 24: Lip synchronization test sequences

Synchronization	**Frame size**	**Frame rate**
Audio lead	QCIF	5 fps
Audio lead	QCIF	10 fps
Audio lead	QCIF	15 fps
Audio lead	SQCIF	5 fps
Audio lead	SQCIF	10 fps
Audio lead	SQCIF	15 fps
Audio lag	QCIF	5 fps
Audio lag	QCIF	10 fps
Audio lag	QCIF	15 fps
Audio lag	SQCIF	5 fps
Audio lag	SQCIF	10 fps
Audio lag	SQCIF	15 fps

The results are reported in Table 25. As expected, an Audio-lead error was easier to detect and neither frame size nor frame rate impacted the detection. The detection of an Audio lag error was 2-3 times more difficult to detect, and both frame size and frame rate had an impact on detection. In the lowest frame rate (5 fps) the motion of the lips was coarse, and it created a feeling that voice and lip motion was not in sync. Some test subjects claimed that even with an error-free sequence, there was a lip sync problem due to a lack of lip motion. A small frame size of SQCIF (Sub Quarter Common Interface Format) made the audio-lag error harder to detect. Some may argue that SQCIF and a frame rate of 5 fps is no longer acceptable with current norms and expectations, and they are correct, but our aim was simply to understand the impact of the frame size and frame rate on lip synchronization error and not to evaluate the experience as a whole.

Table 25: Lip synchronization threshold for a mobile environment

Case	**Audio lag**	**Audio lead**
SQCIF 5 fps	-240 ms	+80 ms
SQCIF 10 fps	-240 ms	+80 ms
SQCIF 15 fps	-280 ms	+80 ms
QCIF 5 fps	-160 ms	+80 ms
QCIF 10 fps	-200 ms	+80 ms
QCIF 15 fps	-200 ms	+80 ms

If only one threshold could be used in some lip synchronization algorithms, then our recommendation would be that the sequence would be in sync between – 200 ms and + 80 ms

6.2. VIDEO QUALITY

The second study developed the idea that with a small screen size, a threshold can be established in which the video quality is so good that people are no longer able to detect any improvements. Sending streamed data beyond this threshold would be a waste of network bandwidth. There are two variables in this threshold (frame rate and bitrate). Also the content of the sequence affects the threshold: a high motion movie scene requires more data than a newscaster sitting still.

Brun et al. aimed to determine the frame-rate limit for low bitrate scenes (below 64 kbps) in [Brun 2004]. According to the test, frame rates of 5 fps and below cannot provide satisfying quality and frame rates higher than 15 fps are not necessary. [Winkler 2006] showed that with low bitrates (max 48 kbps) 8 fps can produce better subjective quality than 15 fps. They used six different video sequences with a different level of motion and in all of them, the 8-fps sequence were rated higher by subjective evaluation. Nemethova et al. also compared bitrates and frame rates with subjective quality [Nemethova 2004] and concluded that in high motion sequences, for example, 32 kbps with 15 fps was subjectively better than 64 kbps with 10 fps. In case of a low-motion sequence, the result was opposite. The study of Superiori et al. [Superiori 2008b] proposed using fewer bits in macroblocks that belong to the frame background. The example of the study was a soccer game where the audience has very little impact on the viewing experience while the players and the ball are the most critical visual elements. The test results clearly show that the viewing experience was improved when more bits were used for the essential portions of the video. Similar attempts to allocate more resources to more critical video segments were also studied in [Korhonen 2010]. The difference from the previous study mentioned was that a decision was made between audio and video. For example, in football, it is more important to receive good quality video than audio, while in a news sequence, it is the opposite. Ranking the importance of video vs. audio in a movie sequences turns out to be more difficult since both media were considered to be important.

Each frame requires a certain amount of data, therefore, an increase in frame rate increases the demand on the bitrate. If the bitrate does not follow the frame rate demand, frames begin to lose details. Table 26 summarizes the quality aspects of different cases.

Table 26: Video quality characteristics

	Low frame rate	**High frame rate**
Low bitrate	Low visual quality Discontinuous motion	Low visual quality Smooth motion
High bitrate	High visual quality Discontinuous motion	High visual quality Smooth motion

The studies mentioned above tested codecs used in mobile devices (H.264), bitrates that are achievable in mobile networks and frame sizes that can be viewed with mobile devices, but the tests were held in a controlled lab environment and using computer screens. In an unpublished study of Miikka Lundan and Igor Curcio (from year 2003), we tested end-users ability to detect difference in video qualities with a mobile device in a natural environment. The codec used in our tests was H.263 Profile 0 Level 10, so the results are not entirely comparable to the above tests (H.264 provides better quality in low bitrates than H.263). In theory, the H-264 bitrate thresholds should be lower, since it can achieve the same quality with lower bitrate. Our initial question was*: Is there a threshold where people no longer detect quality improvement in small screen sizes?* In theory if the bitrate is increased, then it is possible to increase the frame rate and also increase the detail level of each frame (See Figure 7 and Figure 8 in Section 3.3), but in practice the small screen size may make the improvement undetectable to the user, and this is the threshold we examined.

In our test we changed three parameters: frame rate, bitrate and sequence motion (newscaster / movie trailer). Due to video compression techniques high motion requires much more bits than low motion i.e. in a newscaster sequence, two consecutive frames are almost identical, while in an action movie, two consecutive frames may vary a lot. The movie trailer was from a comedy, so it was not containing the highest possible motion like some action films would have. There were 2-4 people (male and female) in the scene. Frames changed from portrait to full-body and some sections had more motion (falling and frantic discussion) while other sections had only a gentle discussion with close-up and low motion (except changes from one speaker to another). In the newscaster sequence there was a portrait of a male newscaster.

There were no scene cuts on the sequence, but there were some text changes (news titles) next to the newscaster.

In our test, we asked users to compare two consecutive sequences that lasted 30 seconds each and tell whether they noticed any improvement in the latter sequence. In order to concentrate on video quality, the sequences were without audio. We enlisted 90 persons in our test, and each person watched a randomly chosen set of video sequences that improved one after another. Improvement was categorized in three levels:

1. Major improvement: More than 50% of the users could detect the difference (Quality improvement from 0.5 to 1.0).
2. Moderate: Quality difference detection was between 30% and 50%. (Quality improvement from 0.3 to 0.5).
3. Minor: Less than 30% of the users could detect the difference. (Quality improvement less than 0.3).

Following figures show the quality improvement in cumulative graph. Each step cannot be more than 1, since it indicates that 100% of the users detected the difference. If the difference between steps is 0, then no improvement has been detected by any of the users. If the improvement becomes negative, then users preferred the first sequence better than the second sequence.

At first, we tested a low-motion sequence with a low frame rate (10-frames per second). We started at 16-kbps sequences, and major improvement was seen up to 64 kbps (80% detection rate on average) and minor improvement until 96 kbps (20%). With high-motion sequence the results were similar: major improvement was reported up until 64 kbps (80%) and moderate up to 96 kbps (40%). Figure 17 summarizes the result.

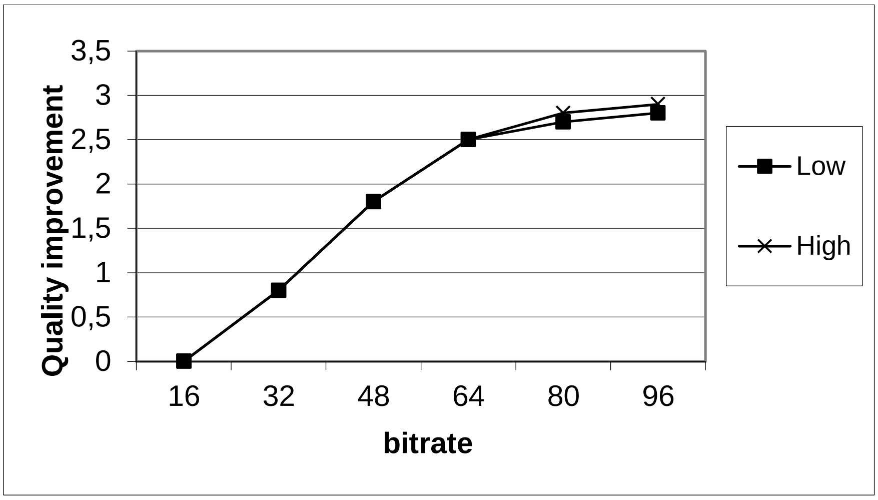


Figure 17: Relative quality of bitrate for low frame rate

Why there was so little difference between the high-motion sequence and the low-motion sequence? After we noticed this phenomenon, we began to ask how the user detected the quality improvement, and it turned out that there was one aspect that we did not consider when planning the tests: the context of the low-motion sequence. The fact that we used a newscaster and not some other low-motion sequence like slowly moving clouds was the key to these results. Users were very sensitive to detect improvement in the face of the newscaster.

With higher frame rate (20 fps) one can find a difference between high-motion sequence and low-motion sequence (Figure 18). There is major improvement up to 64 kbps (60%) and moderate improvement up until 96 kbps (35%) in low-motion sequence. With high-motion sequence, major improvement was also gained up to 64 kbps, but the improvement was more significant (80% detection rate) and minor up to 96 kbps (20%).

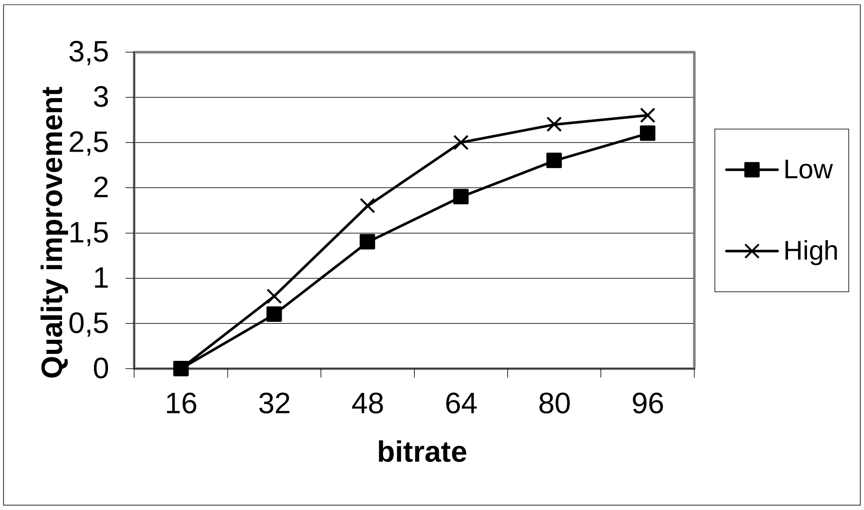


Figure 18: Relative quality of bitrate for high frame rate

The tests with frame rate changes show also the importance of the context in this test (Figure 19). In 32 kbps test cases, the motion change makes no difference and the results show clearly that 10 fps provides best subjective quality and there is not enough data to provide 15 fps motion. In 64 kbps tests, the context starts to have a role. The fact that in low-motion 64 kbps sequence 15 fps provides lower subjective quality that 10 fps, while in high motion 64 kbps sequence the situation is opposite, would not make sense unless we would know the impact of the context. People complained that the details of the face were clearer on 10 fps sequence than in 15 fps sequence and improved motion (smoothness) did not compensated this issue.

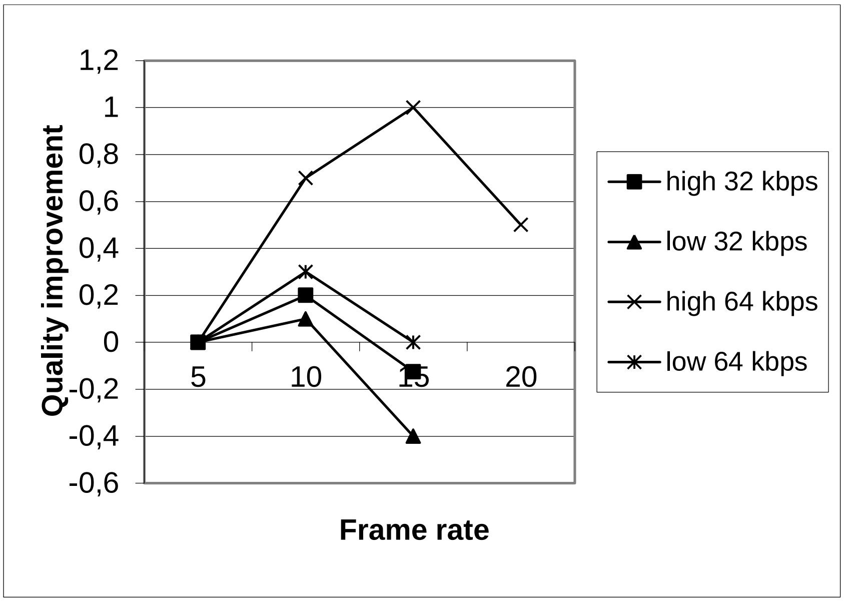


Figure 19: Relative quality of frame rate

Based on these tests, there are two thresholds: If a video service aims to maintain low costs, as well as maintain adequate quality, we recommend that they use 64-kbps sequences, but there is very little quality improvement to gain above 96 kbps. Also 10 fps motion is a good average, but some sequences may benefit of a 15 fps rate. Since this test was done with a mid-range mobile device with 2.1-inch screen, one must note that new high-end devices that have larger displays (such as the Nokia N97 with a 3.5–inch screen) may still gain noticeable improvement with higher bitrates.

Chapter 7

Conclusion

THIS dissertation demonstrated that streaming is possible even in a basic GPRS network. The viewing experience may be moderately fulfilling, due to limited bandwidth, but the idea of having a “TV” in your pocket may be intriguing to many consumers. Increased bandwidth provided by EGPRS and WCDMA naturally improves the media quality, but also in these networks, the basic problems were inherited from the GPRS network.

The term, *non-bitrate-guaranteed networks,* combines a GPRS network and partially EGPRS and WCDMA networks. The main characteristics of these networks are that they cannot guarantee allocated network bandwidth or preserve packet arrival delay bounds (in EGPRS and WCDMA, this depends on the Traffic Class used). With these limitations, the streaming service always has to be prepared for changes in order to provide a meaningful and satisfying experience. A streaming client may change from one network cell to another, and a client may wander in a position where the signal quality grows weak or the cell load changes dramatically.

This dissertation provides a solution to all of these challenges. They are not the only solutions available “on the market”, but the key difference is that they are based fully on a streaming client and server interactions and require no changes to the underlying network.

The cell reselection management method allowed for reporting the data transfer gap caused by cell reselection and hid the data gap from end-user by retransmitting the lost data. Together with the bandwidth adaptation, it also allows preventing buffer underflow.

This bandwidth adaptation method allows a streaming client to accurately report the network bandwidth to the streaming server, and the server can then decide what changes are needed to adapt to the change. This method also reports if a streaming client changes from a non-bitrate-guaranteed network to a bitrate-guaranteed network or vice versa. This information may have an impact on what bitrate adaptation technique is used.

This dissertation also introduced a novel method to report statistical data or errors about the streaming session. The earlier methods (such as RTCP reports) had very limited capabilities to report problems of a streaming session. This new method also allows for negotiating the measured parameters at the beginning of the session and changing them during the session. This allows the method to adapt to different types of streaming sessions or clients.

In addition to these technical improvements, this dissertation also studied two subjective thresholds of streaming. The main focus was to understand, is there a subjective limitation that affect the detectability of technical improvements. Improvements can be theoretical and even detectable by some technical measurement, but in the end it is the human eyes and ears that make the improvement important. Small display size is the main reason for these limitations, but it is compensated by shorter viewing distance. Relative size difference of the display (screen size/viewing distance) is much less than technical size difference to traditional television screen.

In lip-synchronization tests, the detectability of lip-synchronization error was measured. Similar measurements had already been taken with a TV (or computer screen), but the use of a mobile device was a novel approach and provided a new threshold for the mobile device to maintain lip-synchronization.

Another subjective evaluation was organized around video quality with a different bitrate and frame rate. In theory, the video quality increases when the bitrate and frame rate increases, but the small screen size of mobile devices may actually render the improvement undetectable. This dissertation confirmed this theory by finding thresholds after the end-users were no longer able to see any improvement in video quality. For example bitrates provided by WCDMA network go far beyond the needs of mobile device streaming client. The threshold values found are somewhat codec- and display-size specific, and therefore, do not apply to all devices. This dissertation did not attempt to identify correct threshold for all the combinations of codecs and displays, but rather to demonstrate that such thresholds exist. One interesting finding among all these technical details was that the context of the video also affected this threshold. If the video included a close-up of a person speaking, the threshold was higher than in other cases.

Mobile networks improve at a rapid pace, which improves the streaming experience. In a technologically-oriented country like Finland, one easily forgets that the cutting-edge achievements of engineering are not readily available for everyone worldwide and even older technology needs to be improved since it continues to be used widely in many other parts of the world. This dissertation provides tools that could be used by streaming service providers to improve the streaming experience in countries and regions where the mobile network cannot solve the problems examined in this study.

Bibliography

Packet-switched Streaming Service (PSS) specifications

[3GPP-4 26.233] 3GPP TS 26.233 v4.2.0, Transparent end-to-end Packet-switched Streaming Service (PSS); General description (Release 4), March 2002.

[3GPP-4 26.234] 3GPP TS 26.234 v4.5.0, Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release 4), December 2002.

[3GPP-5 22.233] 3GPP TS 22.233 v 5.0.0, Transparent end-to-end Packet-switched Streaming Service; Stage 1 (Release 5), March 2002.

[3GPP-5 26.233] 3GPP TS 26.233 v5.0.0, Transparent end-to-end Packet-switched Streaming Service (PSS); General description (Release 5), March 2002.

[3GPP-5 26.234] 3GPP TS 26.234 v5.7.0, Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release 5), March 2005.

[3GPP-6 22.233] 3GPP TS 22.233 v 6.3.0, Transparent end-to-end Packet-switched Streaming Service; Stage 1 (Release 6), September 2003.

[3GPP-6 26.233] 3GPP TS 26.233 v6.0.0, Transparent end-to-end Packet-switched Streaming Service (PSS); General description (Release 6), September 2004.

[3GPP-6 26.234] 3GPP TS 26.234 v6.13.0, Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release 6), March 2005.

[3GPP-7 22.233] 3GPP TS 22.233 v 7.0.0, Transparent end-to-end Packet-switched Streaming Service; Stage 1 (Release 7), June 2006.

[3GPP-7 26.233] 3GPP TS 26.233 v7.0.0, Transparent end-to-end Packet-switched Streaming Service (PSS); General description (Release 7), June 2007.

[3GPP-7 26.234] 3GPP TS 26.234 v7.5.0, Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release 7), March 2008.

Other specifications and articles

[3GPP 26.071] 3GPP TS 26.071 v 4.0.0, Mandatory speech CODEC speech processing functions; AMR Speech CODEC; General description (Release 4). April 2001.

[3GPP 26.142] 3GPP TS 26.142 v7.3.0, Dynamic and Interactive Multimedia Scenes (DIMS) (Release 7), June 2009.

[3GPP 26.171] 3GPP TS 26.171 v 5.0.0, Adaptive Multi-Rate – Wideband (AMR_WB) speech codec; General description (Release 5), April 2001.

[3GPP 26.290] 3GPP TS 26.290 v 6.3.0, Extended AMR-Wideband codec; Transcoding functions (Release 6), June 2005.

[3GPP 26.401] 3GPP TS 26.401 v 6.2.0, Enhanced aacPlus general audio codec; General description (Release 6), April 2005.

[3GPP VCEG-M77] Varsa et. al., "Common Test Conditions for RTP/IP over 3GPP/3GPP2", Video Coding Experts Group STUDY GROUP 16 Question 6 (Q.6/SG16, VCEG), Thirteenth Meeting: Austin, Texas, USA, 2-4 April, 2001, http://wftp3.itu.int/av-arch/video-site/0104_Aus/VCEG-M77.doc

[3GPP VCEG-N37] Varsa et. al., "Common Test Conditions for RTP/IP over 3GPP/3GPP2 – Software and Amendments", Video Coding Experts Group STUDY GROUP 16 Question 6 (Q.6/SG16, VCEG), Fourteenth Meeting: Santa Barbara, CA, USA, 21-24 September, 2001, http://wftp3.itu.int/av-arch/video-site/0109_San/VCEG-N37.doc

[Alexiou 2006] Alexiou, A. et al., "Equation based congestion control for video transmission over WCDMA networks" 20th International conference on advanced information networking and applications, April 2006.

[Antoniou 2007] Antoniou, P. et al., "ADIVIS: A novel adaptive algorithm for video streaming over the internet", 18th International symposium on personal, indoor and mobile radio communications, September 2007.

[ATSC IS -919] ATSC, "ATSC implementation subcommittee finding: Relative timing of sound and vision for broadcast operations", Doc. IS-191, June 2003.

[Beaufort 2002] Beaufort, D. et al., "Measured performance of TCP friendly rate control protocol over a 2.5G network", 56th IEEE Vehicular technology conference, 2002.

[Bhebhe 2008] Bhebhe, L. & Arjona, A., "Data outage across 3G & 2G wireless network", International Symposium on a world of wireless, mobile and multimedia networks, June 2008.

[Brun 2004] Brun, P. et al., “Subjective assessment of H.264/AVC video for low-bitrate multimedia messaging services”, International conference on image processing. October 2004

[Chemiakina 2003] Chemiakina, S. et al., “QoS enhancement for adaptive streaming services over WCDMA”, IEEE journal on selected areas in communications, Volume 21:Issue 10, December 2003.

[Curcio 2005] Curcio, I.D.D. & Leon, D., “Evolution of 3GPP streaming for improving QoS over mobile network”, IEEE International conference on image processing, September 2005

[Diaz 2007] Diaz, A. et al., “Evaluating video streaming over GPRS/UMTS networks: A practical case”, 65th IEEE Vehicular technology conference, April 2007

[Dogan 2002] Dogan, S. et al., “Error-resilient video transcoding for robust internetwork communications using GPRS”, IEEE transactions on circuits and systems for video technology, June 2002.

[Ericsson 020407] 3GPP TSG-SA WG4 Meeting#22, Tdoc S4-(02)0407, Proposal for bandwidth selection in PSS, July 22-26, 2002, Tampere, Finland. http://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_22/Docs/S4-020407.zip

[GIF 87a] CompuServe Incorporated: "GIF Graphics Interchange Format: A standard defining a mechanism for the storage and transmission of raster-based graphics information", Columbus, OH, USA, 1987.

[GIF 89a] CompuServe Incorporated: "Graphics Interchange Format: Version 89a", Columbus, OH, USA, 1990.

[GSM 02.60 Rel 97] GSM 02.60 v 6.3.1, General Packet Radio Service (GPRS); Service description; Stage 1 (Release 1997), November 2000.

[GSM 03.60 Rel 97] GSM 03.60 v 6.11.0, General Packet Radio Service (GPRS);Service description; Stage 2, (Release 1997) September 2002.

[GSM 03.64 Rel 97] GSM 03.64 v 6.4.0, General Packet Radio Service (GPRS); Overall description of the GPRS radio interface; Stage 2 (Release 1997), November 1999.

[GSM 03.64 Rel 99] GSM 03.64 v 8.12.0, General Packet Radio Service (GPRS); Overall description of the GPRS radio interface; Stage 2 (Release 1999), April 2004.

[GSM 04.60 Rel 99] GSM 04.60 v 8.27.0, General Packet Radio Service (GPRS); Mobile Station (MS) - Base Station System (BSS) interface; Radio Link Control/ Medium Access Control (RLC/MAC) protocol (Release 1999), September 2005.

[GSM 05.09 Rel 99] GSM 05.09 v 8.6.0, Link Adaptation (Release 1999), June 2005.

[GSM 22.100 Rel 99] GSM 22.100 version 3.7.0, UMTS phase 1 (Release 1999), October 2001.

[GSM 22.146 Rel 6] GSM 22.146 version 6.8.0, Multimedia Broadcast/Multicast Service; Stage 1 (Release 6), December 2008.

[GSM 23.205 Rel 4] GSM 23.205 version 4.11.0, Bearer-independent circuit-switched core network; Stage 2 (Release 4), September 2006.

[GSM 23.228 Rel 5] GSM 22.228 version 5.15.0, IP Multimedia Subsystem (IMS); Stage 2 (Release 5), June 2006.

[GSM 25.201 Rel 99] GSM 25.201 version 3.4.0, Physical layer - General description (Release 1999), June 2002.

[GSM 25.933 Rel 5] GSM 25.933 version 5.4.0, IP transport in UTRAN (Release 5), December 2003.

[GSM 44.060 Rel 4] GSM 44.060 version 4.23.0, General Packet Radio Service (GPRS); Mobile Station (MS) - Base Station System (BSS) interface; Radio Link Control/Medium Access Control (RLC/MAC) protocol (Release 4), November 2005.

[GSM 45.092 Rel 6] GSM 45.092 version 6.8.0, Flexible Layer One (FLO) (Release 6). January 2005.

[GSM 45.093 Rel 6] GSM 45.093 version 6.0.1, Feasibility study on Single Antenna Interference Cancellation (SAIC) for GSM networks (Release 6). November 2004.

[Gurtov 2002] Gurtov, A. et al., "Multi-layer protocol tracing in a GPRS network", 56th vehicular technology conference, 2002.

[Hoymann 2002] Hoyman, C. & Stuckmann, P., "On the feasibility of video streaming over GPRS/EGPRS", Global Telecommunications Conference, November 2002.

[Huang 2008] Huang, Z. & Yi, B., "Adaptive transmission for mobile streaming over WCDMA networks", 4th International conference on wireless communications, networking and mobile computing. October 2008.

[Ishibashi 2000] Ishibashi, Y. & Tasaka, S., “A Comparative survey of synchronization algorithms for continuous media in network environments”, 25th IEEE conference on Local Computer Networks, November 2000

[Ishibashi 2001] Ishibashi, Y. et al., “A comparison of media synchronization quality among reactive control schemes”, 20th Joint conference of the IEEE computer and communications societies, April 2001.

[ISO 14496-2] ISO/IEC 14496-2:2004: "Information technology – Coding of audio-visual objects – Part 2: Visual".

[ISO 14496-3] ISO/IEC 14496-3:2005: "Information technology – Coding of audio-visual objects – Part 3: Audio".

[ITU-R 1359] ITU-R BT.1359.1: “Relative timing of sound and vision in broadcasting, January 1998.

[ITU-T H263] ITU-T Recommendation H.263 (01/05): "Video coding for low bit rate communication".

[ITU-T H264] ITU-T Recommendation H.264 (03/05): "Advanced video coding for generic audiovisual services". See also ISO/IEC 14496-10:2005: "Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding".

[ITU-T T81] ITU-T Recommendation T.81 (1992) | ISO/IEC 10918-1:1993: "Information technology – Digital compression and coding of continuous-tone still images – Requirements and guidelines".

[Johnson 2005] Johnson, C. et al., “Inter-system handover for packet switched services”, 6th IEE International Conference on 3G and beyond, November 2005, Washington DC.

[Kang 2005] Kang, S-R. & Loguinov, D., “Impact of FEC overhead on scalable video streaming” International workshop on network and operating system support for digital audio and video (NOSSDAV), June 2005

[Kodikara 2002] Kodikara, C. et al., “Performance improvement for real-time video communication by link adaptation in E-GPRS networks. 3rd International conference on 3G mobile communication technologies, May 2002

[Korhonen 2005] Korhonen, J. et al., “Toward bandwidth-efficient and error-robust audio streaming over lossy packet networks,” Multimedia Systems, Springer-Verlag, vol. 10, no. 10, pp. 402-412, August 2005.

[Korhonen 2008] Korhonen, J. & Frossard, P., “Error control for video streaming with small data units,” Fourth International Mobile Multimedia Communications Conference (MobiMedia’08), Oulu, Finland, July 2008.

[Korhonen 2010] Korhonen, J. et al., “On the relative importance of audio and video in the presence of packet losses,” International Conference on Quality of Multimedia Experience (QoMEX’10), Trondheim, Norway, June 2010.

[LA 2004] LinearAccoustic white paper, “Audio and video synchronization: Defining the problem and implementing solutions”, 2004.

[Liu 2007] Liu, T. et al., “Subjective quality evaluation of decoded video in the presence of packet losses”, IEEE international conference on acoustics, speech and signal processing, April 2007.

[Lundan 2002] Curcio, I.D.D. & Lundan, M., “On RTCP feedback for mobile multimedia applications,” Proc. IEEE International Conference on Networking (ICN02), 26-29 August 2002, Atlanta, Georgia, USA, part 14, pp 637-648.

[Lundan 2004] Curcio, I.D.D & Lundan. M., “Streaming media,” Patent application WO 2004/036824 (April 2004), Granted patents FI116816 (February 2006), TW248740 (February 2006).

[Lundan 2005] Aksu. E., Curcio, I.D.D. & Lundan, M., “Transmission of embedded information relating to a Quality of Service,” Patent application WO 2005/022865 (March 2005), granted patent EP1661366 (May 2006).

[McGinley 2009] McGinley, M. (editor), Urbanization rates of countries, Encyclopedia of earth, May 2009. http://www.eoearth.org/article/Urbanization_Rates_of_Countries,

[MMA-MDLS] Mobile DLS, MMA specification v1.0. RP-41 Los Angeles, CA, USA. 2004.

[MMA-MXF] Mobile XMF content format specification, MMA specification v1.0., RP-42, Los Angeles, CA, USA. 2004.

[Nguyen 2004] Nguyen, H. et al., “Robust and adaptive transmission of compressed video streams over EGPRS” Consumer Communications and Networking Conference, January 2004.

[Nemethova 2004] Nemethova, O. et al., “Subjective evaluation of video quality for H.264 encoded sequences”, Proceeding of SymboTic. October 2004

[Nemethova 2006] Nemethova, O. et al., “An adaptive error concealment mechanism for H.264/AVC encoded low-resolution video streaming”, in Proc. Proceedings of 14th European Processing Conference (EUSIPCO), Florence, Italy, September, 2006

[Nilsson 2003] Nilsson, M.E. et al., "Adaptive multimedia streaming over IP", International conference on visual information engineering, July 2003

[OMADRM2] Open Mobile Alliance: "DRM specification 2.0".

[Rexhepi 2004] Rexhepi, V. et al., "Performance of streaming services in GERAN A/Gb mode" 60th IEEE Vehicular technology conference, September 2004.

[Rexhepi 2005] Rexhepi, V. et al., "Handover of packet-switched service in GERAN A/Gb mode", Global Telecommunications Conference, December 2005.

[RFC 2083] IETF RFC 2083: "PNG (Portable Networks Graphics) specification Version 1.0", Boutell T., et al., March 1997.

[RFC 2326] IETF RFC 2326: "Real Time Streaming Protocol (RTSP)", Schulzrinne H., Rao A. and Lanphier R., April 1998.

[RFC 2354] IETF RFC 2353: "Options to repair of streaming media", Perkins, C. & Hodson, O., June 1998.

[RFC 2429] IETF RFC 2429: "RTP payload format for the 1998 version of ITU-T recommendation H.263 video (H.263+), Burmann C. et al., October 1998

[RFC 2616] IETF RFC 2616: "Hypertext Transfer Protocol – HTTP/1.1", Fielding R. et al., June 1999

[RFC 2733] IETF RFC 2733: "An RTP payload format for generic Forward Error Correction", Rosenberg, J. & Schulzrinne, H., December 1999.

[RFC 3267] IETF RFC 3267: "RTP payload format and file storage format for Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) audio codecs", Sjoberg J. et al., June 2002

[RFC 3448] IETF RFC 3448: "TCP Friendly Rate Control (TFRC): Protocol specification", Handley, M. et al., January 2003.

[RFC 3550] IETF RFC 3550: "RTP: A Transport protocol for real-time applications", Schulzrinne H. et al., July 2003

[RFC 3711] IETF RFC 3711: "The Secure Real-time Transport Protocol (SRTP)", Baugher M. et al, March 2004.

[RFC 4566] IETF RFC 4566: "SDP: Session Description Protocol", Handley M., Jacobson V. and Perkins C., July 2006.

[RFC 4585] IETF RFC 4585: "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", Ott J. et al., July 2006

[RFC 4588] IETF RFC 4588: “RTP retransmission payload format”, Rey, J. et al., July 2006

[RFC 5506] IETF RFC 5506: Support for Reduced-Size Real-Time Transport Control Protocol (RTCP):Opportunities and Consequences, Johansson, I. & Westerlund. M., April 2009.

[Schorr 2004] Schorr, A. et al., “Adaptive media streaming in heterogeneous wireless network”, IEEE 6th workshop on multimedia signal processing, September 2004.

[Solhjoo 2003] Solhjoo N. et al., “Static analysis of inter-system handover”, 4th International conference on 3G mobile communication technologies, June 2003.

[SP-MIDI] Scalable Polyphony MIDI specification version 1.0, RP-34, MIDI Manufacturers Association, Los Angeles, CA, February 2002.

[STD 0006] IETF STD 0006: "User Datagram Protocol", Postel J., August 1980.

[STD 0007] IETF STD 0007: "Transmission Control Protocol", Postel J., September 1981.

[Steinmetz 1996] Steinmetz, R., “Human perception of jitter and media synchronization”, IEEE Journal of Selected Areas in Communications, Vol.14, No.1, January, pp.61-72

[Superiori 2008a] Superiori, L. & Rupp, M., “Smart sorting of H.264/AVC encoded sequences for applications over UMTS networks”, in Proc. Proceedings of IEEE International Symposium on Consumer Electronics, Algarve, April, 2008

[Superiori 2008b] Superiori, L. et al., “Optimization of audience encoding in low-resolution soccer video sequences”, in Proc. Proceedings of Asilomar Conference on Signals, Systems and Computers, Asilomar, California, USA, October, 2008

[W3C CC/PP] W3C recommendation: “Composite Capability / Preference Profiles (CC/PP): Structure and Vocabularies 1.0”, January 2004. http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/,

[W3C RTF] W3C recommendation: “Resource Description Framework (RDF)”, February 2004. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[W3C SVG-Tiny] W3C last call working draft: "Mobile SVG Profile: SVG Tiny, Version 1.2", August 2004. http://www.w3.org/TR/2004/WD-SVGMobile12-20040813/, [W3C SMIL] W3C recommendation: "Synchronized Multimedia Integration Language (SMIL 2.0)-[Second Edition]", January 2005. http://www.w3.org/TR/2005/REC-SMIL2-20050107/

[WAG UAProf] Wireles Application Group: "User Agent Profile specification, November 1999.

[Winkler 2006] Winkler, S. & Faller, C., "Perceived audiovisual quality of low-bitrate multimedia content" IEEE transactions on multimedia, Volume 8, Issue 5, October 2006.

[XHTML] W3C Recommendation: "XHTML Basic", December 2000. http://www.w3.org/TR/2000/REC-xhtml-basic-20001219

Specification web pages

3GPP PSS, GPRS, EGPRS and WCDMA related specifications http://www.3gpp.org/specifications

IETF RFC RFC-specification: http://rfc-editor.org/rfcsearch.html

ITU-T http://www.itu.int/rec/T-REC-T/e

W3C http://www.w3c.org

OMA http://www.openmobilealliance.org/

Publications

[P1] **Miikka Lundan** and Igor Curcio, "3GPP streaming over GPRS Rel. '97," *Proc. 12th IEEE International Conference on Computer, Communications and Networks (*ICCCN '03), 20-22 October 2003, Dallas, TX, U.S.A., pp. 101-106.

[P2] Igor Curcio and **Miikka Lundan,** “Optimal 3GPP Packet-switched Streaming Service (PSS) over GPRS networks,” *Multimedia tools and application*, Volume 35, Number 3, December 2007.

[P3] **Miikka Lundan,** "Streaming over EGPRS," *Proc. IEEE 9th Symposium on Computers and Communications* (ISCC04), 28 June - 2 July 2004, Alexandria, Egypt, pp. 969-974.

[P4] **Miikka Lundan** and Igor Curcio, "Mobile streaming service in WCDMA network," Proc. IEEE 10th Symposium on Computers and Communications (ISCC05), 27-30 June 2005, Cartaghena, Spain, pp. 231-236.

[P5] Igor Curcio and **Miikka Lundan**, "Event-driven RTCP Feedback for Mobile Multimedia Applications," *Proc. 3rd Finnish IEEE Wireless Communications Workshop* (FWCW'02), 29 May 2002, Helsinki, Finland, pp 1-2.

[P6] Igor Curcio, Juha Kalliokulju and **Miikka Lundan**, "AMR Mode Selection Enhancement in 3G Networks," *Multimedia Tools and Application*, Volume 28, Number 3, March 2006.

[P7] EmreAksu, Igor Curcio,David Leon, **Miikka Lundan,** Viktor Varsa, and Ru-Shang Wang, "Bandwidth adaptation,"*Patent application WO2004/028095 (April 2004), Granted patents FI116498 (November 2005), CN1314247 (May 2007), EG23825 (September 2007) and HK1084268 (October 2007).*

[P8] Igor Curcio and **Miikka Lundan**, "Human Perception of Lip Synchronization in Mobile Devices," *Proc. IEEE Symposium on a World of Wireless, Mobile and Multimedia Networks* (WOWMOM 2007), 18-21 June 2007, Helsinki, Finland.