Streaming Video in Wireless Networks

Service and Technique

Fredrik Montelius
Oscar Larsson

LITH-ISY-EX-3227

Linköping 2001-12-11

Streaming Video in Wireless Networks

Service and Technique

Master Thesis

Linköping Department of Electrical Engineering

Fredrik Montelius

Oscar Larsson

LiTH-ISY-EX-3227

Supervisor: Marcus Antonsson, Anna Englund, Accenture AB
Jörgen Ahlberg, Linköping University
Examiner: Robert Forchheimer, Linköping University

Avdelning, Institution
Division, department

Institutionen för Systemteknik
581 83 Linköping

Datum
Date
2001-12-11

Språk
Language

❑Svenska/Swedish
☒ Engelska/English

❑ _______________

Rapporttyp
Report: category

❑ Licentiatavhandling
☒ Examensarbete
❑ C-uppsats
❑ D-uppsats
❑ Övrig rapport
❑ _______________

URL för elektronisk version

http://www.ep.liu.se/exjobb/isy/2002/3227/

ISBN

ISRN LiTH-ISY-EX-3227-2002

Serietitel och serienummer
Title of series, numbering

ISSN

Title: Streaming Video in Wireless Networks - Service and Technique

Titel: Strömmande video i trådlösa nätverk – tjänst och teknik

Författare
Author Fredrik Montelius, Oscar Larsson

Sammanfattning
Abstract
The purpose of this thesis is to present an attractive service for the third generation mobile network that includes streaming video. A prototype application for this service is to be built. The technique behind streaming video is to be presented so that it comes clear what kind of problems and solutions that are associated with streaming. Finally, a platform for streaming video is to be tested and evaluated through different channels.

The attractive service presented in this thesis is MMS - Multimedia Messaging Service. Today's popular SMS is evolving beyond text to multimedia. Multimedia will be part of the next generation messaging service called MMS, which will, in an advanced shape, include streaming video. MMS is expected to be a successful service for the next generation's cell phones.

The WAP Forum and the 3GPP industry groups are responsible for standardizing MMS. The standard defines an MMS architecture, which has a number of key elements that interact with each other. The prototype application that was built is called mVideo Messaging and is an MMS that is built on the basis of the MMS standard. The kernel of the prototype is a platform from PacketVideo that makes it possible to stream video over wireless networks.

Theories and tests makes it clear that the parameters affecting the video quality can be found at the source/target while coding and compressing, as well as at the streaming-channel. At the channel there are above all three network problems - packet loss, end-to-end delay and delay jitter. To deal with these matters, new protocols have been developed. At the source/target it is important use an efficient compression scheme. MPEG-4 is a new compression scheme that suits very well for streaming video through wireless channels. MPEG-4 make use of scalability, is object oriented, and is optimized for streaming between 9,6 kbps and 4 Mbps.

The service proposed in this thesis as a future service for 3G is practicable. It is also shown that the service can be built using the technology of today.

Nyckelord
Keywords

Streaming video, MPEG, MPEG-4, MMS, EMS, SMS, Multimedia Message Service, Image Coding

Abstract

The purpose of this thesis is to present an attractive service for the third generation mobile network that includes streaming video. A prototype application for this service is to be built. The technique behind streaming video is to be presented so that it comes clear what kind of problems and solutions that are associated with streaming. Finally, a platform for streaming video is to be tested and evaluated through different channels.

The attractive service presented is called MMS - Multimedia Messaging Service. Today's popular SMS is evolving beyond text to multimedia. Multimedia will be part of the next generation messaging service called MMS, which will, in an advanced shape, include streaming video. MMS is expected to be a successful service for the next generation's cell phones.

The WAP Forum and the 3GPP industry groups are responsible for standardizing MMS. The standard defines an MMS architecture, which has a number of key elements that interact with each other. The prototype application that was built is called mVideo Messaging and is an MMS that is built on the basis of the MMS standard. The kernel of the prototype is a platform from PacketVideo that makes it possible to stream video over wireless networks.

Theories and tests makes it clear that the parameters affecting the video quality can be found at the source/target while coding and compressing, as well as at the streaming-channel. At the channel there are above all three network problems - packet loss, end-to-end delay and delay jitter. To deal with these matters, new protocols have been developed. At the source/target it is important use an efficient compression scheme. MPEG-4 is a new compression scheme that suits very well for streaming video through wireless channels. MPEG-4 make use of scalability, is object oriented, and is optimized for streaming between 9,6 kbps and 4 Mbps.

The service proposed in this thesis, a future service for 3G is practicable. It is also shown that the service can be built using the technology of today.

Preface

This master thesis project has been performed at Accenture AB in Stockholm, Sweden. It has been a great opportunity for us to involve ourselves in an interesting project and increase our skills and knowledge in the area. We would like to take the opportunity to thank all the people that have given us help and support during the work.

To mention a few, we would like to thank Joakim Sjöman and Eric Matsgård for believing in our ideas and us, and giving us the opportunity of performing our master thesis at Accenture AB. Many thanks to our supervisors at Accenture AB, Marcus Antonsson and Anna Englund, and our project leader Richard MacDowall for giving us good support and advises during the whole project. We also would like to thank other staff at Accenture that have been directly involved in the project. Especially Henrik Dahlgren, Zak Keith, and Karin Backman for their help with developing the graphical interface for the prototype, Mårten Barkwall for creative ideas during early stages of the project and Paul D Godwin for checking the grammar.

Furthermore we would like to give a great thank to our examiner Robert Forchheimer and our academic supervisor Jörgen Ahlberg at Linköping University, Sweden for continuous feedback and insightful discussions. A thanks also goes out to Hassan Nasreddin, Linköping University for interesting discussions.

We also cannot forget the staff at PacketVideo, especially Kevin Carrol, Szu-wei Wang, Alan Hebert. Thanks to you for providing us with proper information and for keeping us on the right track during the work.

Finally we would like to thank everyone else that have shown interest in the project and given us valuable comments, suggestions, support or help.

Linköping

December 2001

Table of contents

1 Introduction

This report constitutes a master thesis for the degree Master of Science in Engineering – Information Technology, and Master of Science in Engineering – Computer Science and Engineering at the Linköping University, Sweden.

1.1 Background

The first generation (1G) of mobile cellular communications systems were analog and primarily used for voice. They were introduced in the late 1970s and early 1980s. Starting in the 1990s, second generation (2G) systems used digital encoding and include GSM. Except for GSM's SMS text message service, 2G systems have been used mostly for voice. Between now and the third generation (3G), which is expected in the 2003-2005 time frame, a variety of 2G+, or 2.5G, techniques are being employed to improve the speed of data for enhanced e-mail and Internet access. These technologies include packet enhancements for GSM (GPRS) and improved data rates for GSM (EDGE). 3G is designed for high-speed multimedia data and voice. Its goals include high-quality audio and video.

Over the last few years there has also been a dramatic improvement in the quality of IP-based network media technologies. Both real time and on-demand media can now be created, served and played at PCs. In order to play smoothly, video data needs to be available continuously and in the proper sequence without interruption. Until fairly recently, it had to be downloaded in its entirety to the PC before it could be played. With streaming, the file remains on the server.

High-speed networks and improved IP-based technologies create new conditions for future mobile video services.

1.2 Purpose

The purpose of this thesis has been divided into four rather separate goals.

1. To present an attractive service for 3G that contains streaming video.
2. To build a prototype application for this service, using a suitable streaming video platform.
3. To present problems and technical solutions that are associated with video streaming.
4. To study and evaluate the chosen streaming video platform through different channels.

1.3 Reader's guide

The report is divided into four major parts (chapter 2, 3, 4 and 5). Every part deals with each goal respectively and in the same order as in section 1.2. At the end of these four chapters conclusions are drawn in order to answer the question at issue. These four chapters can be read individually although reading chapter 4 before 5 is strongly recommended.

Chapter 2 starts off with looking at the new conditions that arises when the world switches from the old type of mobile networks (2G) to the new ones (2.5G[1] and 3G). The chapter then continues with the evolution from text to multimedia in messaging services. Finally, an overview of the MMS standard model is given. This model is used as a theoretical reference in chapter 3.

In chapter 3 the prototype developed is described. The chapter starts with stating what technologies that are to be used and then, the application is described both on a high and a low level. At the end of the chapter some recommendations for further work with the system are given.

Chapter 4 presents problems and technical solutions for streaming video. This chapter is rather theoretical and eases the understanding of chapter 5.

Chapter 5 describes different experiments that have been made in order to verify some of the theories in chapter 4 and also to evaluate the streaming video platform.

In chapter 6 a summary of the whole thesis is given.

The references used are a variant of the Harvard system. Because of the many Internet-references, a number that refer to the reference-list at the end of the document is used instead of author and year of publication. If a chapter is based on one source only, only one reference is given at the end of the chapter.

A list of acronyms is available at the end of the document. All acronyms are normally explained when used for the first time. The reader is assumed to be familiar with basic concepts used in computer science.

1.4 Method

The work has been divided into four main stages: One prestudy, two iterations of application development and finally an evaluation/ documentation stage.

For the first goal several brainstorming sessions with experts from different areas of interest was held. These brainstorm sessions ended up in 15 application alternatives. Together with experts from Accenture AB the four

[1] 2.5G or 2G+, includes a variety of techniques that are being employed to improve the speed of data for enhanced e-mail and Internet access. These technologies include packet enhancements for GSM (GPRS) and improved data rates for GSM and TDMA (EDGE).

most attractive alternatives were picked out. Together with partners at Accenture AB the one to build was chosen.

In order to fulfill the second goal an official standard for how to build the service was studied. This standard was then modified in order to meet necessary demands. For the third goal classical theory studies was conducted. The fourth goal was taken care of by setting up a test-environment that made it possible to emulate different channels and network situations. The test methodology and all results have been performed by the authors only and were not endorsed by PacketVideo. Neither did PacketVideo participate in the testing.

During the early stages time not invested in scheduled activities like interviews and brainstorm sessions were used for studies of the programming environment. Documentation of the work was partly being created continuously in association with the work and partly after the application development was complete.

2 Multimedia Messaging Service

This chapter deals with the first issue for the master thesis – to present an attractive service for 3G that contains streaming video.

The first step is to carry out a survey about the prerequisites for 3G network services. Next, the market for these services are studied and evaluated in order to find out how the conditions for services change in 3G networks compared to 2G.

To find a new successful service for 3G the successful services in the 2G – the SMS is studied. The SMS is evolving beyond text by the path to EMS[2] and MMS[3]. MMS will in an advanced shape include streaming video, which is an important demand of the service.

In the final part of this chapter, a theoretical model of how to build this service is presented.

2.1 The wireless multimedia landscape

According to the Wireless Multimedia Forum (see below), wireless networks will be ready to begin supporting content-rich applications such as streaming media at the end of 2001.

One reason for the urgency of getting more content in the networks, is that the development of multimedia content for delivery over traditional wired Internet connections is on a collision course with the maturation of the mobile market, according to WMF. Mobile devices will complement traditional PCs and might even displace them as the preferred Internet access mechanism. The users will then demand access to the same services they enjoy today on their desktop computers via their handheld devices. In addition, mobile users will seek a host of new services aimed directly at their mobile needs, such as location-based information services. These industry developments are setting the stage for the mass delivery of rich multimedia content over mobile networks.

For many years, standards fragmentation has stunted progress in the mobile industry at the air link protocol layer and in the Internet industry with multiple competing streaming media standards. To prevent fragmentation from stunting the growth of the wireless multimedia market, a group of 56 industry players[4] called the Wireless Multimedia Forum (WMF) has begun specifying technologies that they will use to achieve interoperability among wireless multimedia products and content throughout the marketplace. Their

[2] Enhanced Messaging Service, more described in chapter 2.3.2.
[3] Multimedia Messaging Service, more described in chapter 2.3.3.
[4] To mention a few of the charter members: Cisco Systems, Emblaze Systems, NTT DoCoMo, PacketVideo, Sonera. In [44] all members are listed.

intent is to cultivate the widespread delivery of new content-rich services across IP[5]-based wireless networks by enabling products and services from many players to work together. This work will result in new sources of income for content developers and service providers. The work will also provide new application choices for the user community. All this work accelerates the marketplace for wireless multimedia content delivery. Both for today's 2G wireless networks, which run at speeds below 64 kbps[6], and for emerging higher-speed 2.5G and 3G networks.[43]

2.2 Creation of new markets

This section answers a few questions that are vital when a new market is created. What is made in order to create good conditions for the new market? Who are the players? Who are the users? What pricing models can be used?

The section is describing the beginning of a new mobile market from the Wireless Multimedia Forum's (WMF) point of view. The major part of section 2.2 reflects the content of *The Business Case for Wireless Multimedia Services* [45], which is a report from WMF.

2.2.1 The conditions are changing

The conclusion of section 2.1 is that the mobile market is ready for new revenue opportunities. This involves content and application service developers, network service providers, makers of network infrastructure equipment and mobile communication device manufacturers. The industry participants are working hard in order to realize the true business potential of content-rich new services for a mobile customer base.

One important condition the wireless industry is working with, is to ensure the interoperability of the basic technologies used in developing and delivering multimedia network services. Other significant industry activities are to bring new faster mobile networks and new pricing models for these.

Until recently, mobile capabilities in many parts of the world have been largely limited to voice capabilities and low-speed, data-only sessions. Because the services for these networks were originally developed to support voice phone calls, many current pricing models are based on per-minute connect times. For data and multimedia applications, though, connect time pricing models have reduced user enthusiasm for new services. In such a pricing model the slower the network, the more the user pays.

These are some of the factors that have until recently inhibited rich data- and multimedia-over-wireless applications.

[5] Internet Protocol, implements the network layer of the protocol, which contains a network address and is used to route a message to a different network or subnetwork.
[6] KiloBits Per Second

The WMF is coordinating the efforts of vendors and worldwide standards bodies to prevent the emergence of "islands" of applications and networks without interoperability. As a result, the group has made available early recommendations of common technologies for use in wireless multimedia networks. The goal is to promote the use of compatible compression, file format, transport, and other technologies. This will allow applications and services developed using common interfaces to run over any service provider's network infrastructure and to be delivered to any type of end-user mobile device. If these tasks could be solved with technologies used in common by the industry players, the wireless multimedia suppliers would get to the market faster with rich multimedia content services that can reach a broad range of customers.

The mobile telephony-based digital networks now in use throughout the world are predominantly circuit-switched and run at about 20 Kbps speeds. In addition, wireless network operators have begun building higher-speed, packet-switched mobile 2.5G and 3G networks. These networks will better accommodate the new generation of content-rich applications by providing greater bandwidth, eliminating the need to dial network sessions and, in some cases, eliminating the connect time pricing model.

Both 2.5G and 3G networks provide "always-on," LAN-like connections to the users. Packet-switched services are better for the users from a performance and pricing standpoint. This changes the conditions for the pricing models. A better model from the user point of view is one where they pay for usage, that is, for transmitting or receiving content, not by connect time. Therefore, they can remain connected at no charge and, for example, receive email as it arrives, rather than having to frequently dial up to check for messages. [45]

2.2.2 The players in the new arena

In order for the market opportunity to be fully realized for all members of the wireless multimedia supply chain; a base level of common technologies must be in use, end to end, to ensure a broad, interoperable network reach. The wireless multimedia supply chain includes:

- Content developers/providers
- Network operators and service providers
- Makers of network core- and edge-equipment
- Manufacturers of mobile terminals such as cell phones, personal digital assistants and pagers.

The combination of richer content, faster networks and the ability to reach a large group of users with different wireless devices opens up new value-added service opportunities for content providers. In addition, network operators worldwide that have spent billions of Euros (see figure 2.1) on wireless licenses for delivering next-generation services need to quickly begin earning revenues on their investments. They are, thus, seeking new value-added content and application services to deliver to their end customers.

At the same time the industry is working hard to ease development costs, prevent market fragmentation and shorten all players' time to market with new services. [45]

Country	UMTS licenses	Total license cost (in billions)	License cost per capita
Austria	6	€0.7	€85
Germany	6	€50.5	€611
Italy	5	€12.2	€213
Netherlands	5	€2.7	€169
UK	5	€36.8	€630
Finland	4	€0.0	€1 [7]
Norway	4	€0.2	€40
Sweden	4	€0.0	€0.0
Spain	4	€0.5	€13

Figure 2.1: Worldwide network operators have spent billions on 3G spectrum licenses and need to quickly begin recouping their investments. [45]

On 16th December 2000, Sweden awarded UMTS[8] third-generation (3G) mobile phone network licenses to four consortia.

The Swedish Post and Telecom Authority (PTS) said two of the four licenses on offer went to domestic operators Europolitan, controlled by Britain's Vodafone Group Plc and Tele2, the cellular arm of Netcom, in cooperation with Societé Europeène de Communication SA (SEC), a consortium grouping Orange, the mobile subsidiary of France Telecom, BredbandMobil, a joint venture between Bredbandsbolaget and Internet consultancy Framtidsfabriken, Skanska AB, NTL Ltd and Schibsted ASA also won a license. The fourth was given to a consortium called HI3G Access AB made up of Investor AB and Hutchison Whampoa Ltd.

Sweden awarded its UMTS licenses based on evaluations of the applicants' financial strength, technical feasibility, roll-out speed and commitment to geographical coverage. Ten groups representing a total of 30 companies applied. [42]

[7] Due to the small population of Finland (approximately 5 millions), the total license cost falls out of region of the statistics.

[8] Universal Mobile Telecommunications System, the European implementation of the 3G wireless phone system.

2.2.3 The user situation

According to Ericsson, revenues from 3G will come from two main sources: charging for network access and charging for content and value-added services or transactions. Today, mobile operators are in possession of one of the most valuable parts of the value chain for any information society, namely the distribution, which currently includes "ownership" of the customers in the form of mobile subscribers. [22]

Network operators in certain parts of the world where markets are culturally ready for value-added wireless services are especially eager to deliver new content services. In Japan, for example, the penetration of cell phones is higher than that of PCs. This is one reason behind the success of NTT DoCoMo’s i-mode[9] service. In Japan, i-mode is the default service for Internet, email, online banking, ticket purchasing and peer-to-peer electronic game playing and video messaging. The i-mode service has, according to DoCoMo, a user base of more than 23 million. DoCoMo expects that figure to rise to about 30 million by March 2002. [45]

Consumers, particularly in Europe and Asia, are already accustomed to the everyday use of mobile devices for basic communication. In Finland, the penetrations of the cell phones and PCs are approximately equivalent, so it is necessary to have a strategy for reaching both sets of users. Finnish mobile operator Sonera, for example, is in trials with content management company Worldzap for delivering live and near-live video clips of sports events and entertainment to mobile devices. The service is stated to be launched at the 2002 Soccer World Cup. [45]

The United States faces its own challenges. Here, PC deployment far outweighs cell phone deployment. In the U.S., a cell phone is considered by many consumers to be an emergency device, rather than a primary communications and productivity tool. Part of the reason is that the fees for landline network services are far less expensive than wireless service rates; the opposite is true in some other parts of the world. So challenges in stimulating the U.S. market are that content must be compelling, pricing models must be palatable, and performance must be similar to what users have experienced with landline PC connections. [45]

2.3 The evolution from text to multimedia

As seen above, many companies believe there is a demand for new services that make use of the conditions in next generation’s mobile network. In this section the successful short message service is described to see how and why it will evolve to multimedia message service. This section is based on *Next Messaging, An Introduction to SMS, EMS and MMS* [27] and almost all facts come from that report.

[9] A packet-based information service for mobile phones from NTT DoCoMo (Japan). I-mode was the first smart phone system for web browsing and grew very quickly after its introduction in 1999.

SMS is the ability to send and receive text messages to and from mobile telephones. EMS is the ability to send a combination of simple melodies, pictures, sounds, animations, modified text and standard text as an integrated message for display on an EMS compliant handset. MMS is the ability to send and receive messages comprising a combination of text, sounds, images and video to and from MMS capable handsets.

2.3.1 Short Message Service (SMS)

The first short message was sent in December 1992 from a PC to a mobile phone on the Vodafone GSM[10] network in the UK. Messages that uses SMS are short (100-200 characters), and involves sending text only messages between phones or computers. Today's SMS has several advantages inherent in its fundamental features. One of these is the possibility to store and forward messages. This means in the case that the recipient is not available, the short message is stored at an SMS Center. When the recipient later becomes available the message is delivered. Another great advantage is confirmation of delivery. This means that the user knows that the short message has arrived. In the Circuit Switched Data environment, there is an end-to-end connection and therefore the user has knowledge that a connection has been established and the data is being transferred. SMS has been designed to take the burden of message delivery and delivery verification away from the user through features such as store and forward and confirmation of delivery.

But today's SMS has also several disadvantages. The unit short message is currently limited to 140 octets[11]. It would be preferable to have a length that is several times this size. The structure of the SMS Protocol Data Unit as defined in the GSM 03.40 standard is inflexible. The Data Coding Scheme, Origination Address, Protocol Identifier and other header fields are fixed. This has constrained the number of possible scenarios that can be indicated when developing applications.

There is no doubting the success of the SMS. By August 2000, nine billion SMS messages were being sent each month, including six billion in Europe alone. This growth is predicted by Mobile Lifestreams to continue growing until 2004 at least since the mobile phones, infrastructure, specifications, market development and awareness are in place today. Over time, as users connect to networks that offer more advanced data services such as the GPRS[12] and buy mobile terminals that support them, they will use those new services for new and existing applications. [27]

[10] Global System for Mobile Communications. A digital cellular phone technology that is the predominant system in Europe, but is also used around the world.
[11] An eight-bit storage unit. In the international telecommunication community, octet is often used instead of byte.
[12] General Packet Radio Services

2.3.2 Enhanced Messaging Service (EMS)

EMS is the ability to send a combination of simple melodies, pictures, sounds, animations, modified text and standard text as an integrated message for display on an EMS compliant handset. EMS is an enhancement to SMS but is very similar to SMS in terms of using the store and forward SMS Centers, the signaling channel and the like to realize EMS. [27]

The EMS came about as a submission by Ericsson to the standards committees. Ericsson presented the structure of EMS to the ETSI/3GPP[13] committees and stated that they would only commit more resources to propagating EMS if the handset manufacturers all committed to supporting it. All of the major handset vendors with the exception of Nokia did, in 2001, commit to support the concept of EMS. Because of this, the EMS standards have evolved and are now stable and complete as part of the 3GPP technical specification *"3G TS 23.040, Technical realization of the Short Message Service (SMS)"* ([2]). [27]

EMS supports not just plain text but left, center or right alignments of text, normal, large or small font sizes and normal, bold, italic, underlined or strikethrough text. EMS supports three picture formats[14], which all are plain black and white, there is no gray scale for shading. EMS will support moving pictures in two sizes[15]. The standards define a number of animations relating to sadness, flirtatiousness, gladness, skepticism and grief. These predefined animations are stored on the individual handset so they are not sent over the air. [27]

Enhanced Messaging Service makes use of the long established and widely used User Data Header (UDH) common in SMS. In SMS, the UDH makes it possible to include binary data in a short message prior to the text message itself. EMS has little or no impact on today's SMS Centers. The introduction of EMS should be totally transparent to SMS Centers since they already support the User Data Header. The principal modification to existing SMS Centers would be in the case that mobile network operators wanted to charge differently for EMS. In such a case, the SMS Center would have to record the relevant technical values and generate call detail records accordingly. [27]

The potential for EMS in practice will depend upon the availability of EMS compliant handsets and the support by manufacturers such as Nokia. Awareness amongst potential users of EMS and availability of compelling content will of course also be critical to EMS' success. [27]

---

[13] 3rd Generation Partnership Project is a cooperation of standards organizations (ARIB, CWTS, ETSI, T1, TTA and TTC) throughout the world that is developing the technical specifications for IMT-2000. 3GPP develops the W-CDMA technology, and 3GPP2 develop the cdma2000 technology, all of which increase data rates for 3G wireless communications.

[14] Small (16 by 16 pixels), large (32 by 32 pixels) or variable sized pictures. The standards recommend a maximum picture size of 96 by 64 pixels- but this depends upon the handset vendor's implementation.

[15] Small (8 by 8 pixels) and large (16 by 16 pixels).

2.3.3 Multimedia Messaging Service (MMS)

MMS is the ability to send and receive messages containing a combination of text, sounds, images and video to and from MMS capable handsets. MMS will in an advanced shape include streaming video.

The trends for the growth in MMS have their roots in the changes that are taking place at all levels within the mobile Internet. Enabling bearers such as GPRS, EDGE[16] and 3G are becoming available. Enabling technologies such as Bluetooth, WAP[17], MExE[18] and SyncML[19] are all initiatives that support this new direction toward the Mobile Internet. An interesting aspect of these new technologies is the shift in attitudes of the companies involved; from competition to co-operation for the greater good of the industry.

The Multimedia Messaging Service is according to the 3GPP standards [1], [3] "a new service which has no direct equivalent in the previous ETSI/GSM world or in the fixed network world."

In order to enable the MMS, 3GPP introduces new messaging platforms to the mobile networks. These platforms are called MMS Relay, MMS Server, MMS User Databases and new WAP Gateways. In section 2.4 these platforms are described in more detail. MMS will require not only new network infrastructure but also new MMS compliant terminals since MMS will not be compatible with old terminals. This means that before it can be widely used, MMS terminals must reach a certain penetration. Although MMS is a service standardized by the 3GPP, it can also be offered on GPRS networks. [27]

The 3GPP MMS standard defines that the User Agent shall provide the following application layer functionalities: [3]

- Multimedia Message composition
- Retrieval of Multimedia Messages (initiate delivery to the User Agent).
- Presentation of notifications to the user
- Multimedia Message presentation

The user agent may of course provide more functionality such as decryption/encryption or message storing on device, but the operations above must be supported. [3]

---

[16] Enhanced Data rates for Global Evolution. An enhancement to the GSM and TDMA wireless communications systems that increase data throughput to 384 Kbps.

[17] Wireless Application Protocol. A standard for providing handheld devices with secure access to e-mail and Web pages.

[18] Mobile Execution Environment. The MExE specification details a flexible and secure application environment for 3G (and 2G+) mobile devices.

[19] SyncML is the common language for synchronizing all devices and applications over any network. The SyncML Initiative is sponsored by Ericsson, IBM, Lotus, Matsushita, Motorola, Nokia, Openwave, Starfish Software and Symbian.

2.3.4 Comparison between SMS and MMS

Having looked at the features of SMS and now introduced new features of MMS, a comparison between these two are given in this section. SMS and MMS have some similarities and some discontinuities, as detailed below:

Store and Forward. Both SMS and MMS are non-real time services. This means that there is an intermediate platform such as the SMS Center and the MMS Relay that the short or multimedia messages pass through.

Confirmation of message delivery. Another characteristic that SMS and MMS hold in common is the fact that both include confirmation of message delivery. The sender of the message can find out whether or not the message they sent was successfully delivered.

Communications Type. The communications type is also likely to be similar in SMS and MMS. Most messages are thought to be person-to-person communication. The types of communication will simply be less textual and more visual. People will still be communicating with other people, often one on one, but the form of this communication will be multimedia with MMS rather than text with SMS.

Media supported. The SMS supports text and binary as media. However, the overwhelming majority of all SMS messages are pure and plain text. Multimedia messages on the other hand can be coded in various media from text to images to sounds to video clips to a combination of these. As such, the MMS is a much more powerful service that supports far more media and rich media.

Delivery mechanism. All short messages are sent and received over the signaling channel. That is a channel that is an additional transport mechanism on GSM networks over and above the traffic channels themselves.

Multimedia messages will be transmitted over the traffic channels themselves where other data types from voice to data will also be transported. The high capacity of 3G networks will mean that all these different traffic types can share the same radio resource without the likelihood of congestion.

Protocols. When SMS was standardized in the early to mid-1990s, the Internet was an academic communications medium. The original ETSI specifications for SMS closely mandated some areas of SMS and left others open to competition. As a result, proprietary protocols were developed; every SMS Center vendor developed its own interface such that application developers needed to implement different interfaces when porting their applications and services to network operators that had different SMS Centers.

MMS came of age in the Internet world where open systems and standard protocols reign and a wide range of these protocols exist. MMS uses standard Internet protocols such as MIME[20] and SMTP[21] for access to the MMS Environment.

Platforms. In SMS, the SMS Center is the heart of the service, with all short messages of any type passing through an SMS Center to and from mobile phones. With MMS there are several key platforms within the MMS Environment; including the MMS Relay, the MMS Store, the MMS User Database and other platforms including the existing platforms such as the SMS Center, voice mail platforms plus more.

The comparison is summarized in figure 2.2.

Feature	**SMS**	**MMS**
Store and Forward (non real time)	Yes	Yes
Confirmation of message delivery	Yes	Yes
Communications Type	Person to person	Person to person
Media supported	Text plus binary	Multiple- Text, voice, video
Delivery mechanism	Signaling channel	Data traffic channel
Protocols	SMS specific e.g. SMPP	General Internet e.g. MIME, SMTP
Platforms	SMS Center	MMS Relay plus others

Figure 2.2: Comparison between SMS and MMS [27]

MMS takes a lot of the winning features of SMS but improves and extends those capabilities with better bandwidth and improved ways of accessing and delivering those services. In section 2.4, we will take a look at this MMS architecture. But first we present a timescale for the introduction of this new service. [27]

2.3.5 MMS timescale

To introduce a new service, there is a number of stages before it is established. MMS service developments include standardization, infrastructure development, network trials, contracts placed, network rollout, availability of terminals, application development, and so on. As the development of MMS much is connected to the development of 3G-

[20] Multipurpose Internet Mail Extensions. A common method for transmitting non-text files via Internet e-mail.
[21] Simple Mail Transfer Protocol. The standard e-mail protocol on the Internet.

networks, this time-scale reflects a lot of the stages to the launch of these networks. The stages are shown in figure 2.3 below. [27]

Date	**Milestone**
Throughout 1999	3G radio interface standardization took place, and initial 3G live technical demonstrations of infrastructure and concept terminals shown.
2000	Continuing 3G and MMS standardization with network architectures, terminal requirements and detailed standards.
2000	Governments around Europe and Asia award 3G licenses for phase 1 spectrum.
2001	3G licenses continued to be issued. 3G trials and integration commence.
2001	3G launched in Japan (by NTT DoCoMo and others).
Summer of 2001	First commercial deployment of 3G services become available in Europe.
Start of 2002	Basic 3G capable terminals begin to be available in commercial quantities.
Throughout 2002	-Network operators launch 3G services commercially and roll out 3G. -Vertical market and executive 3G early adopters begin using 3G regularly for nonvoice mobile communications
2002/2003	New 3G specific applications, greater network capacity solutions, more capable terminals become available, fuelling 3G usage.
2004	3G will have arrived commercially and reached critical mass in both corporate and consumer sectors.

Figure 2.3: Service developments for MMS will include standardization, infrastructure development, network trials, contracts placed, network roll out, availability of terminals, application development, and so on. These stages for MMS are shown in the table above [27]

2.4 The MMS architecture

To enhance messaging to this new level a standard was required. The WAP Forum and the 3GPP 3G industry groups are responsible for standardizing MMS. [27]

2.4.1 The WMF end-to-end system model

This section reflects the model that is described in WMF's Recommended Technical Framework Document [44]. The end-to-end model is a technical recommendation to enable interoperability between the different units in it. The recommendations will only be made for three different units - *Content Creation Subsystem, Multimedia Distribution Subsystem and Wireless Multimedia Terminal*, in the system, and the related communication protocols.

The following diagram describes the end-to-end system model for supporting Streaming Multimedia (SMM[22]) and Downloading Multimedia (DMM[23]) applications over wireless systems, i.e. not necessarily MMS-applications.

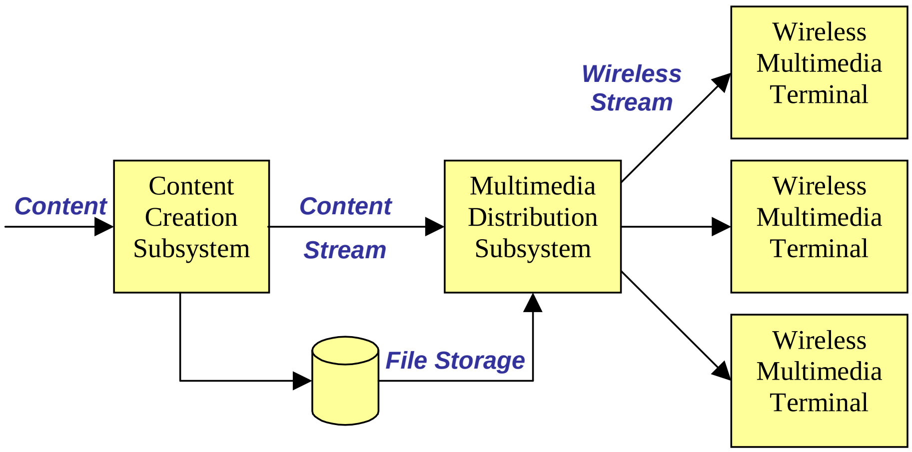


Figure 2.4: The WMF End-To-End SMM and DMM System Model[44]

As shown in figure 2.4, the content is created off-line or in real-time by the content creation subsystem, delivered by the multimedia distribution subsystem, and consumed or stored by the wireless terminal.

Content Creation Subsystem. The content creation subsystem is responsible for taking raw or compressed media content[24], which is stored in a file or captured in real-time, converting it to a content stream that is suitable for delivery, and sending it to the multimedia distribution subsystem. The content stream can also be stored in a file storage[25], from which the multimedia distribution subsystem reads the content stream and distributes it to the wireless multimedia terminal. [44]

---

[22] Streaming Multimedia
[23] Downloading Multimedia
[24] GSM-AMR speech codec shall be supported, MPEG-4 Visual Simple Profile@Level 0 codec shall be supported. ITU-T H.263 baseline shall also be supported.
[25] The MPEG-4 File Format shall be used for transferring of the stored content between content creation subsystem and multimedia distribution subsystem.

Multimedia Distribution Subsystem. The multimedia distribution subsystem receives the multimedia content from the content creation subsystem and streams the content to wireless multimedia terminals, live or with a delay. The multimedia distribution subsystem can also stream pre-stored content to different terminals and also manipulate and/or re-purpose the content. [44]

Wireless Multimedia Terminal. The wireless multimedia terminal receives the multimedia content from the multimedia distribution system, and displays it to the user. The content may be either live or on-demand streaming media for the SMM application, or a downloaded multimedia file for the DMM application. [44]

2.4.2 3GPP MMS model

This section go deeper into the MMS model and architecture defined by the 3GPP in *Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional description; Stage 2* [3]. First a generalized description of the MMS architecture is given, and then the single elements are descibed.

Figure 2.5 shows a generalized view of the Multimedia Message Service architecture. As concluded in previous sections the service shall be integrated in a combination of different networks and networks type. The terminal operates with the Multimedia Messaging Service Environment, MMSE. This environment may comprise 2G and 3G networks, 3G networks with islands of coverage within a 2G network and roamed networks. The MMSE provides all necessary service elements, e.g. delivery, storage and notification functionality. These service elements may be located within one network or distributed across several networks or network types.

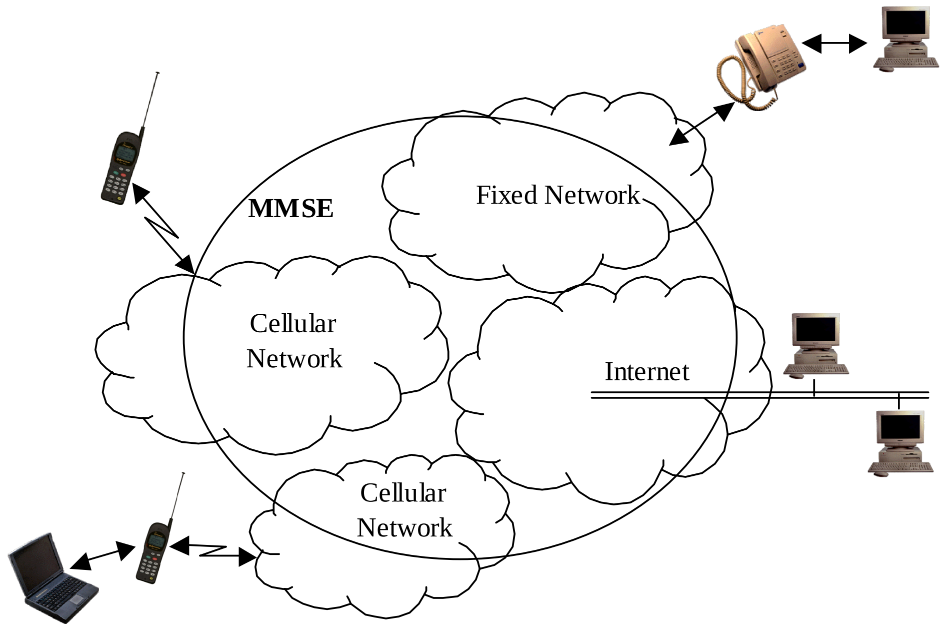


Figure 2.5: 3GPP's general view of MMS provision within different networks [3]

A critical condition for the MMS is just the fact that it has to be integrated in different networks and network types. Therefore the 3GPP's MMS standard is using the Internet protocol and its associated set of messaging protocols. [3]

The Multimedia Messaging architecture has a number of key elements that have been defined and incorporated into this MMS Environment. These key elements are:

- MMS Relay
- MMS Server (or servers)
- MMS Store (or stores)
- MMS User Agent
- MMS User Databases.

The MMS relay is the engine of the MMS and is responsible for the transfer of messages between different messaging systems. The MMS server is responsible for storage and handling of incoming and outgoing messages. The MMS User databases may consist of lots of different data including user profile database, subscription database and information for mobility management. The MMS User Agent is an application layer function that provides the user with the ability to view, compose and handles Multimedia Messages (e.g. sending, receiving, deleting of Multimedia Messages). [27]

Figure 2.6 shows how these different elements will be interacting in the MMS environment. As figure 2.6 implies, it is the MMS Relay that is responsible for the transfer of messages between different messaging systems. But depending on business model, the MMS Server and the MMS Relay may be combined, separate or distributed across different domains. [3]

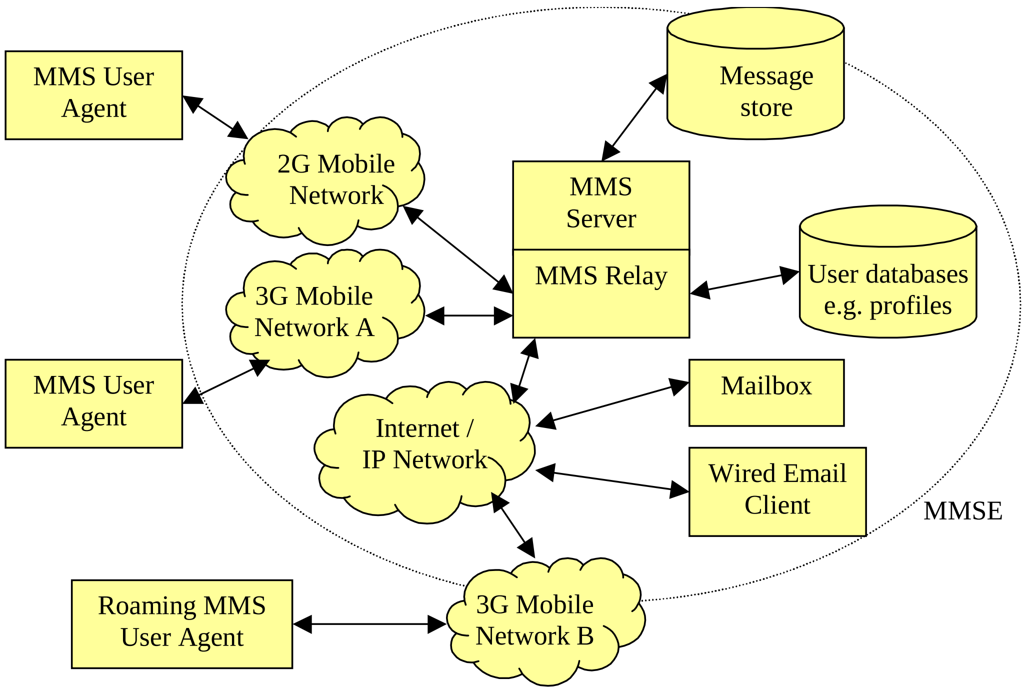


Figure 2.6: MMS Architectural Elements [3]

The MMS Relay should also be able to generate charging data when receiving or delivering Multimedia Messages to the User Agent or another MMSE. The MMS Server is responsible for storage and handling of incoming and outgoing messages and in the User databases information about the MMS Users is stored. In figure 2.6, the 3G Mobile Network B is considered to be part of a separate MMSE. [3]

2.5 Conclusions

Chapter 2 deals with the first issue for the master thesis – to present an attractive service for 3G that contains streaming video. It must also be possible to build a prototype of the service within the master thesis.

The mobile telephony-based digital networks now in use are predominantly circuit-switched. In addition, wireless network operators have begun building higher-speed, packet-switched mobile 2.5G and 3G networks. These networks will better accommodate the new generation of content-rich applications by providing greater bandwidth, eliminating the need to dial network sessions and, in some cases, eliminating the connect time pricing model.

One requirement on the service to present was that it had to involve streaming video. The very popular SMS is evolving beyond text to multimedia. Multimedia will be part of the next generation messaging service called MMS – Multimedia Messaging Service, and will in an advanced shape include streaming video.

To enhance messaging to this new level, a standard is required. The WAP Forum and the 3GPP 3G industry groups are responsible for standardizing MMS. The standard defines an MMS architecture, which has a number of key elements that interact with each other. These are called MMS Relay, MMS Server, MMS Store, MMS User Databases, and MMS User Agent.

MMS is an attractive service for 3G networks and this is the service the prototype should realize.

3 mVideo Messaging

This chapter deals with the second issue for the master thesis – to build a prototype application for the service presented in previous chapter. The prototype shall be using a suitable streaming video platform.

Since one of the requirements (recall section 1.2) for the service was to include streaming video, the service is called mVideo Messaging Service in order to set the focus on the fact that it is a mobile message service containing streaming video.

Since the heart of the service will be the streaming video platform, we start to look at available streaming video platforms in section 3.1. Another important ingredient in the prototype is the handheld. Therefore a small market analysis is undertaken in section 3.2 to see what opportunities today's handhelds give for the mVideo Messaging prototype. After decided which platform and handheld to use, the development of the software begins. The mVideo Messaging application is then presented in section 3.3.

3.1 Streaming video platforms

A streaming video platform makes it possible to send video over Internet to a wireless device such as a cell phone. In this section two different streaming video platforms are investigated and compared. Two may not be the great selection of products, but since this type of product is quite new, the choice of producers is limited. Under investigation are PacketVideo’s PVPlatform and Emblaze Systems Wireless Media Platform.

Since both PacketVideo and Emblaze are members of the Wireless Multimedia Forum (WMF) their platforms are designed according to the WMF end-to-end system model described in section 2.4.1.

3.1.1 Emblaze Systems

Emblaze Systems provides streaming video solutions over Wireless and IP networks. Emblaze offers an end-to-end solution designed to allow live and pre-recorded video and audio content to be viewed on mobile devices. Emblaze Systems is a charter member in the WMF and is also a member of the 3GPP and the MPEG-4 (see section 4.2) Forums. [13]

The Emblaze platform is called Wireless Media Platform and is divided into three parts, Emblaze Encoder, Emblaze Server, and Emblaze Wireless Player. Figure 3.1 shows where in the WMF’s end-to-end system model the element belongs.

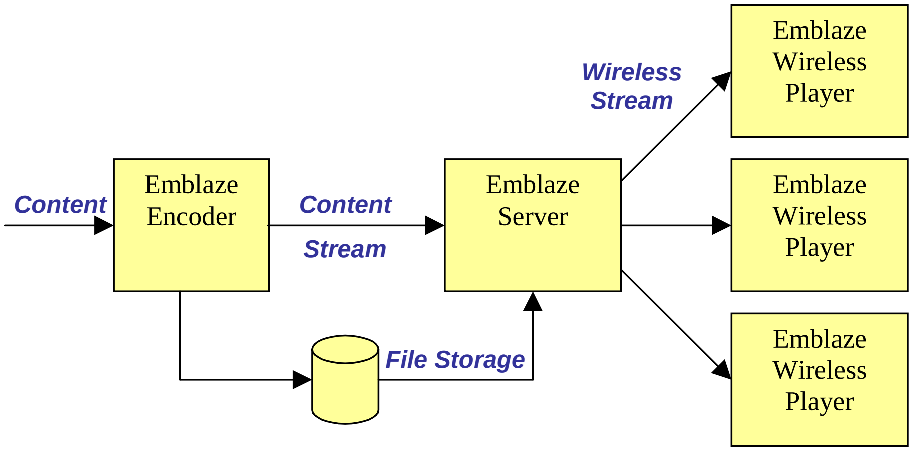


Figure 3.1: The Emblaze Systems Wireless Media Platform

The Emblaze Encoder receives the input video feed and is in charge of compressing and transforming the source into MPEG-4-compliant media [16]. The Emblaze Server is the core of the platform, responsible for receiving, managing and streaming the encoded video content [20]. The content is received from the Emblaze Encoder either as a live stream or as a media file for storage [20]. The Emblaze Server then stores the content in the file storage unit or streams it directly to the Emblaze Wireless Player on hand-held devices [19]. The Wireless Media Platform supports multiple network transport protocols (HTTP[26], RTP[27]) [18].

Emblaze Encoder. The encoder consists of four components - a Video Switch, a Live Encoder, an On-demand Encoder and a Video Mail Recorder.

The Video Switch allows the management of several input video channels to be dynamically directed to Emblaze video encoders (Live / On-demand). The Live Encoder accepts one input video channel, encodes the video and transmits a real-time video stream of the encoded content. The encoded data is transferred to the Emblaze Server to be streamed to the wireless devices via the Streaming Servers. The live data can also be transmitted to any standard web server for Internet-based live video broadcasting.

The On-demand Encoder accepts one input video channel, encodes the video and stores it in an MPEG-4 format file. The MPEG-4 file is then transferred from the encoder's temporary storage and installed in the Emblaze Server storage. The content is then available for streaming to the wireless devices via the Streaming Servers. The MPEG-4 file can also be placed on any standard web server for Internet-based video viewing. The Video Mail Recorder lets a user create and send video email messages. The user needs a standard PC with an Internet connection and a web-cam to record his/hers

[26] HyperText Transport Protocol, the communications protocol used to connect to servers on the World Wide Web.
[27] Realtime Transport Protocol, an IP protocol that supports realtime transmission of voice and video.

video message. The user accesses a web page that supplies them with the video mail recording software. By sending the message to a regular email address, the recorded message can be sent to both wireless video users and Internet PC users. [16]

Emblaze Server. The server consists of five parts - a Network Switch, a Network Load Balancer, a Streaming Server, a Database Server and a Video Storage Filer.

The network Switch connects all the devices in the system and enables each component in the system access to all other components. The Network Load Balancer enables the system to scale its service linearly and to offer high availability by automatically overriding faulty streaming servers "on-the-fly". The Network Load Balancing allows a network service to be known by a single IP address and support the service using multiple servers (with multiple IP addresses). The load-balancing device forwards each request it receives to one of the assigned streaming servers and makes sure that all the assigned servers are equally loaded in order to allow the best quality of service. The Database Server holds all records of the system and video content. The Emblaze Server uses Oracle 8i as a database engine and Sun Solaris 2.7 as the OS platform. [20]

Emblaze Wireless Player. The player is thin-client software that enables mobile users to view on-demand and live MPEG-4 video content and messages on hand-held devices. The player utilizes the pre-installed device browser to browse the available content that is located on the Media Platform. The player can also utilize the email inbox application on the device to receive and display streaming video messages. Emblaze Wireless Players run on the Pocket PC, Win CE and EPOC operating systems (March 2001). [19]

Emblaze Technology. The need to implement video CODECs[28] in hardware components is derived from the limitations of mobile handsets. Handsets carry limited computing power, batteries and memory space. Emblaze currently offers two generations of ASIC[29] solutions (dedicated video processors) leading to less usage of the device resources. [17]

Emblaze A2Plus is a multimedia decoding chip which enables the playback of video on current generation mobile handsets. A2Plus is based on the ARM[30] platform as a high-performance, fully programmable core processor,

---

[28] Compressor/Decompressor, hardware or software that compresses digital data into a smaller binary format than the original.
[29] Application Specific Integrated Circuit, a chip that is custom designed for a specific application rather than a general-purpose chip such as a microprocessor.
[30] ARM chips, a family of RISC-based microprocessors and microcontrollers from ARM Inc., Los Gatos, CA, (www.arm.com).

and includes on-chip memory. The A2 chip includes interfaces to the LCD[31] controller, Voice/Audio codec, and flash memory. A2Plus receives the compressed video and audio from the baseband chip, decodes the video and audio, and sends them to the LCD controller and voice codec/audio D/A converter respectively. The external Flash memory is used for storing A2Plus software. A2Plus includes all memory needed for processing and rendering data, thus eliminating the need for an external memory and saving both power and space that are crucial in mobile applications.[14]

Emblaze A3 is based on the ARM platform, with the addition of video capture and display hardware blocks for enhancing performance and quality, and reducing power consumption of the total solution. Emblaze A3 consists of two main paths: a decode path and an encode path. In the decode path A3 receives the compressed audio/video bit stream from the baseband chip or from the Media storage device, decodes and sends the video to the LCD controller and the audio to the voice codec/audio DAC. In the encode path A3 gets the video from the camera and the voice from the voice codec, encodes the A/V stream and sends them to the baseband chip for transmission or to the Media flash for storage. In a full duplex videophone, the two paths are active together. An external Flash memory is used for storing A3 software and an optional Media flash is used to store the audio/video-compressed data. A3 includes all memory needed for processing and sending data. [15]

3.1.2 PacketVideo

PacketVideo presents another platform for streaming video over Wireless and IP networks. PacketVideo's software platform is an end-to-end solution designed to allow live and pre-recorded video and audio content to be viewed on mobile devices. PacketVideo is a charter member in the WMF and is also a member of the 3GPP and the MPEG-4 Forums. [32]

The PacketVideo platform is divided into three parts, PVAuthor, PVServer, and PVPlayer. Figure 3.2 shows where in the WMF’s end-to-end system model the element belongs.

[31] Liquid Crystal Display, a display technology.

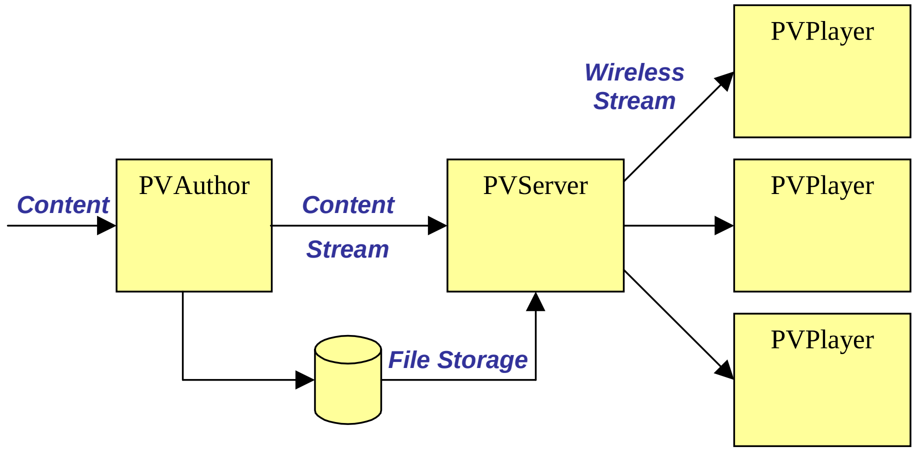


Figure 3.2: The PacketVideo Platform

The PVAuthor receives the input video feed and is in charge of compressing and transforming the source into MPEG-4-compliant media [36]. The PVServer is the core of the platform, responsible for receiving, managing and streaming the encoded video content [35]. The content is received from the PVAuthor either as a live stream or as a media file for storage [35]. The PVServer then stores the content in the file storage unit or streams it directly to the PVPlayer on hand-held devices [35]. A video clip encoded with PacketVideo technology is scalable to different bit-rates and broadcast environments without re-encoding [31].

PVAuthor. PVAuthor is an authoring toolkit that allows for the encoding of streaming material to the MPEG-4 format and supports variable-bit-rate encoding, allowing for the creation of a single encoded stream that can be provided in multiple bandwidth scenarios. [36]

The PVAuthor allows for delivery across a wide array of wireless networks at bandwidths from as low as 9.6 kbps to more than 384 kbps [37]. It provides encoding for live streaming and archiving of content, and enables the delivery of MPEG-4 standards-compliant content to a variety of target devices [37]. The input file can be of the format MPEG-1, MP3, WAV[32], AVI[33], JPEG or BMP[34] [37]. PVAuthor supports Windows 98 SE, 2000, or NT 4.0 (SP 5 & 6a) [39].

PVServer. The PVServer is the heart of the PacketVideo multimedia solution. It consists of three parts: A streaming module, a web- and applications module, and a standard database module.

[32] The native digital audio format in Windows.
[33] Audio Video Interleaved, a Windows multimedia video format from Microsoft.
[34] BitMaP, a Windows and OS/2 bitmapped graphics file format.

The major function of the streaming module is the streaming of multimedia content via UDP[35] to a wireless client. The content can either be stored in MP4 format or it can originate from a live feed from PVAuthor. The primary function of the Web & Application modules is to provide a platform that supports both basic and enhanced core services such as subscriber provisioning, subscriber authorization, and event logging. The Database module provides for the dynamic storage and retrieval of information for the infrastructure services listed above (provisioning, authorization, etc.). [35]

PVPlayer. PVPlayer 2.0 decodes video and audio multimedia using MPEG-4 video decompression, and GSM-AMR audio decompression. The player supports WinCE 3.0, PocketPC, Windows 98, NT4 and Windows 2000 operating systems (March 2001). [35]

PacketVideo Technology. PacketVideo has two techniques which use the advantages of the MPEG-4 file format: FrameTrack and SignalTrack technology

PacketVideo's FrameTrack provides accommodation for variations in available network bandwidth. PVPlayer will report network conditions to PVServer using RTCP[36]. In addition PVServer will adjust streaming rates to accommodate changes via the QoS[37] setting in the PVServer configuration file. [35] More information regarding FrameTrack can be found in section 4.2.6.

PacketVideo SignalTrack error resilience technology helps ensure the video quality by detecting and concealing errors inherent in wireless networks. [38] More information regarding SignalTrack can be found in section 4.2.6.

[35] User Datagram Protocol, a protocol within the TCP/IP protocol suite that is used in place of TCP when a reliable delivery is not required.
[36] Real Time Control Protocol, see section 4.1.4 for more information.
[37] Quality of Service

3.1.3 Emblaze Systems vs. PacketVideo

Platform / Feature	Emblaze Systems	PacketVideo
Basic function	To deliver streaming video to handhelds.	To deliver streaming video to handhelds.
Decoding standard	MPEG-4	MPEG-4
Member of WMF	Yes	Yes
Solution for variations in network bandwidth	No	Yes[38]
Basic design	Encoder - Server - Player	Encoder - Server - Player
Hardware Solutions	Yes[39]	No
Max. number of simultaneous streams	N/A	1000 at 64 kbps [39]
Company relations	None	Good[40]

Figure 3.3: Comparison between platforms.

Figure 3.3 shows a comparison between the two platforms. The PacketVideo platform was chosen on account of several reasons. Two of the most important reasons are the FrameTrack (see section 3.1.2 – PacketVideo Technology) and the company relations between Accenture AB and PacketVideo.

The FrameTrack technology handles variations in available network bandwidth by letting the video player continuously report network conditions to the streaming server and in addition the server will adjust streaming rates to accommodate changes. This is suitable for the mVideo Messaging application, as the networks concerned are networks with varying bandwidths.

The fact that Accenture AB and PacketVideo have signed a Non-Disclosure Agreement (NDA[41]) has been a matter of vital importance in the decision-

[38] FrameTrack (see section 3.1.2 – PacketVideo Technology)
[39] A2Plus and A3 multimedia decoding chips (see section 3.1.1 – Emblaze Technology)
[40] Accenture AB and PacketVideo have signed a Non-Disclosure Agreement (NDA) between them.
[41] An agreement signed between two parties that have to disclose confidential information to each other in order to do business.

making. The use of PacketVideo was strongly recommended by Accenture AB. By that we are not saying that without this NDA our decision would have been different. With this cooperation we can have access to proper technical documentation, good support and assistance.

3.2 Choice of handheld

This section deals with how the handheld for the mVideo Messaging application was chosen.

3.2.1 Market analysis

Below are the demands for the handheld device to use presented:

- **The device should be representative for a future 3G-telephone.** This due to the fact that the mVideo Messaging application is supposed to realize a 3G-service.
- **The device should have a color screen.** Since the platform supports color, this is a feature good to use.
- **It should be possible to send streaming video to the device.** Otherwise the application would be impossible to build.
- **It should be possible to have access to the device in the near future**. This due to the lack of development time.
- **It should be possible to develop an application for the device.** Naturally. Makes a demand on OS.

The handhelds have been evaluated according to the prerequisites presented above.

Device Feature	Casio E-125 Cassiopeia	Compaq iPAQ H3600 series	Ericsson T68	PC-EPhone
1. Look				
2. Display	240 x 320 TFT, 65536 colors	240 x 320 TFT, 4096 colors	256 colors	640 x 480 TFT, 256 colors
3. Remote connection	Yes[42]	Yes[43]	Yes[44]	Yes[45]
4. OS	Windows Pocket PC	Windows Pocket PC	Non-standard [48]	Windows CE
4. Available (March 2001)	Yes	Yes	No[46]	No
5. Development possibilities	eVB, C++[47]	EVB, C++	N/A[48]	eVT[49]

Figure 3.4: Comparison between handhelds

3.2.2 The iPAQ is chosen

The Compaq iPAQ was chosen for the application because it satisfies the demands presented in section 3.2.1. The device has a color screen. Together with a jacket concept that allows for network cards, Compaq iPAQ allows for remote connection. The Compaq iPAQ is easy to get hold of, as opposed to the Ericsson T68 and the PC-EPhone (April 2001). A development environment exists for the OS on the iPAQ.

[42] CompactFlash modem card, Serial Modem, CompactFlash wireless phone connector card, or CompactFlash LAN card [5]
[43] The iPAQ can be equipped with a jacket concept that allows to plug in network cards, e.g. for remote connection. [7]
[44] The T68 can connect through GPRS and Bluetooth [21]
[45] The PC – Ephone has a chip for CDMA mobile communication and a slot for CompactFlash cards.
[46] The launch is planned to the fourth quarter of 2001
[47] eVB = eMbedded Visual Basic
[48] Since the OS is of an unknown standard, no information on development possiblities is available.
[49] eMbedded Visual Tools [26]

Apart from these fulfilled demands, we were also able to access other people's experiences and knowledge of working with the Compaq iPAQ.

Throughout the rest of the report the term "iPAQ" will be used instead of "Compaq iPAQ Pocket PC".

3.3 The mVideo Messaging application

Here the mVideo Messaging application is described. In the first section the main functionality for the application is established. The next section provides a high-level description for the whole prototype and the remaining sections explain the application in detail.

3.3.1 Demand on prototype

The mVideo Messaging application shall provide the same basic functionality as defined by the 3GPP's MMS standard in [3] (see section 2.3.3):

1. The application shall support composition of a Multimedia Message. In our case this means that it has to be able to record a video message and send it to a recipient.
2. The application shall be able to retrieve Multimedia Messages. As the standard states the application only needs to initiate the video stream.
3. The application shall be able to show a "new-message-has-arrived-notification" for the user.
4. The application shall be able to display the Multimedia Message for the recipient.

Apart from the demands derived from the 3GPP's MMS standard we had some demands in addition:

5. The system shall be built on the PacketVideo streaming video platform[50].
6. The application shall be developed for the iPAQ.

3.3.2 High-level overview

In this section we make sure that the established demands in previous sections are fulfilled. A high-level design for the prototype is presented and high-level design decisions and simplifications are clarified in this section.

One demand on the prototype is that it should originate with the MMS architecture. Figure 3.5 shows the MMS architecture with simplifications for the prototype crossed out. The motives for these simplifications are described below.

[50] The platform is described in section 3.1.2.

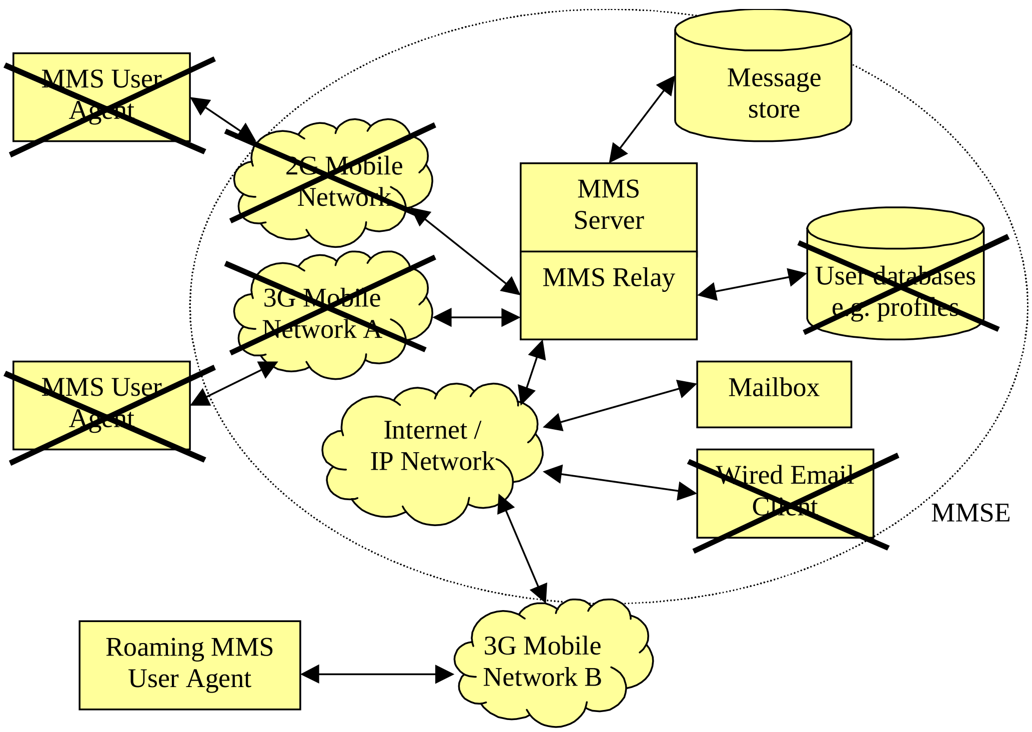


Figure 3.5: The 3GPP defined MMS Architectural Elements with simplifications for the mVideo Messaging Service crossed out

In the MMS architecture the MMS relay is the engine of the system and is responsible for the transfering of messages between different messaging systems. With respect to the purpose of the prototype we decide not to integrate 2G, 2.5G or 3G network in the MMSE. This decision means that we need a very simple MMS Relay, only connected to Internet. Note that this does not imply that we will not be able to stream video over these kinds of networks. By using a separate, for example a 3G-, network as part of a separate MMSE we could stream over the Internet to this network. This scenario is shown in figure 3.5 as 3G Mobile Network B.

The MMS server is responsible for storage and handling of incoming and outgoing messages. The MMS server is fundamental and necessary for the prototype.

The MMS User databases contains data about users. This information could for example be what kind of agent and device a specific user has. Since there will only be one kind of agent and device in the prototype a user database is not necessary.

The MMS User Agent is an application layer function that provides the user with the ability to view, compose and handle Multimedia Messages. The user agent is fundamental and necessary for the prototype.

The reason for keeping the mailbox will be clarified later in this section.

These high-level design decisions create a new simplified MMS architecture - the mVideo Messaging Service architecture. Since all mobile networks with a connection to the Internet will have access to the MMS Relay and MMS Server we will be able to run the prototype in 2G, 2.5G and 3G networks. The mVideo Messaging architecture is shown in figure 3.6.

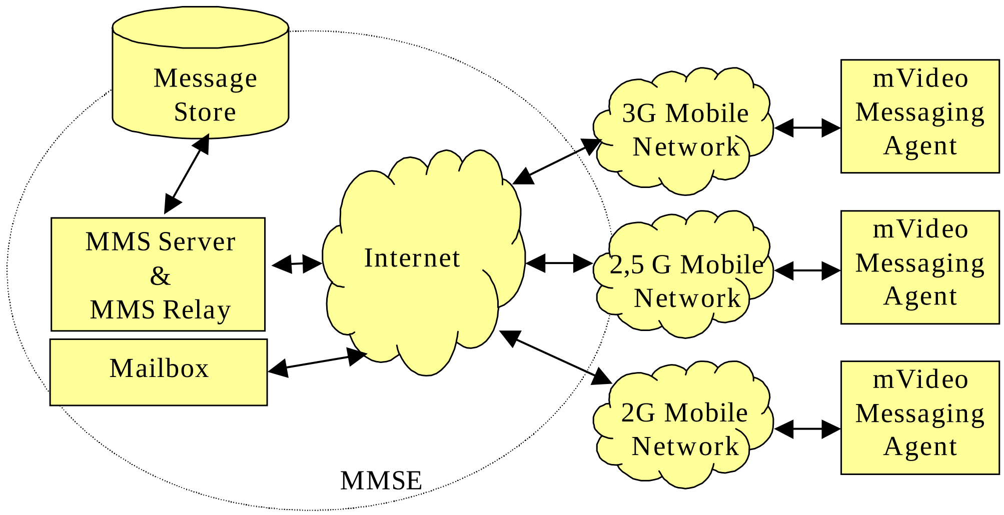


Figure 3.6: mVideo Messaging Service Architectural

The prototype is to be based upon the PacketVideo platform[51]. Therefore the high-level design has to be compatible with this platform, so let us take a look at how the PVPlatform could be integrated in the mVideo Messaging Service architecture.

The PacketVideo platform works (simplified) like this (see figure 3.7): PVAuthor takes a video signal as input and creates a compressed and encoded MPEG-4 file as output or streams the output to PVServer directly. In this prototype the MPEG-4 file alternative is suitable. After this file has been transferred to the File Storage unit the PVServer can stream the content to a PVPlayer that displays the video message.

[51] The PacketVideo platform is described in detail in section 3.1.2.

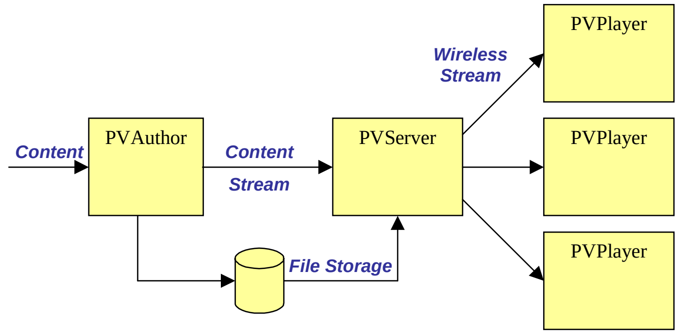


Figure 3.7: The PacketVideo platform

Demand number one for the prototype is to be able to create a video message and send it to a recipient. There are many possibilities of where in the system the compression and encoding of the video signal could be done. It is obvious that the content, a video signal, should come from a camera that is connected to the user agent (e.g. the iPAQ, which should represent a 3G cell phone). But it is not obvious where in the system the PVAuthor should be. We keep in mind that PVAuthor has to run on a PC.

One possible scenario is to connect the video camera to the iPAQ. This means that the uncompressed video signal has to be sent to the PC with PVAuthor in some way. A typical uncompressed message file is about 50 MB while the same message file compressed, will be about 100 KB. If the camera is connected to the iPAQ a lot of data has to be sent to the PC. For that reason we want the encoder to be part of the agent since about 50 MB otherwise has to be transferred over Internet, from the agent to the server. That would be a waste of time and money[52].

Demand number one also includes the possibility to send the message to a recipient. This demand will be discussed together with demand two and three below. But first we just conclude that the PVServer is the heart of the system and becomes part of the MMS Server and MMS Relay.

Demand number two say that the application should be able to retrieve multimedia messages, but the application only needs to initiate the video stream. Demand number three means that the application shall be able to show a "new-message-has-arrived-notification" for the user. These two demands open up the possibility to just send over data that makes it possible to initiate the video stream and show a notification for the recipient. This goes hand in hand with the PacketVideo platform since the compressed and encoded video files should be stored on the File Store unit from which PVServer could stream the video to PVPlayer. The conclusion is that we do not want to send the encoded file to the recipient but just transfer it to the file

[52] Example, it would take about 8 minutes to transfer the uncompressed video signal/file to the encoder if the user has a 100 kbps connection to the encoder.

store unit. But to be able to initiate the video stream and to show a notification some information must be sent to the recipient.

This information could for example be sent with SMS, ordinary email or with some other method. The MMS standard however only covers SMS or/and email (see section 2.3.4 – Protocols). Since the purpose of the prototype is to build a service for the next generation's mobile network we want to take advance of the new possibilities in those. One of the most significant changes is the possibility to be connected to the Internet and only pay for the amount of data that is sent and not for the connection duration[53]. This means that an email-client for example, could run as a background service on an agent and check for new email at a low cost. With these arguments we decide to use ordinary email to send the information to the recipient.

Demand number four says that the application should be able to display the multimedia message for the recipient. This is done with the PVPlayer, which has to be a part of the agent.

The simplifications above create an mVideo Messaging-sending model shown in figure 3.8.

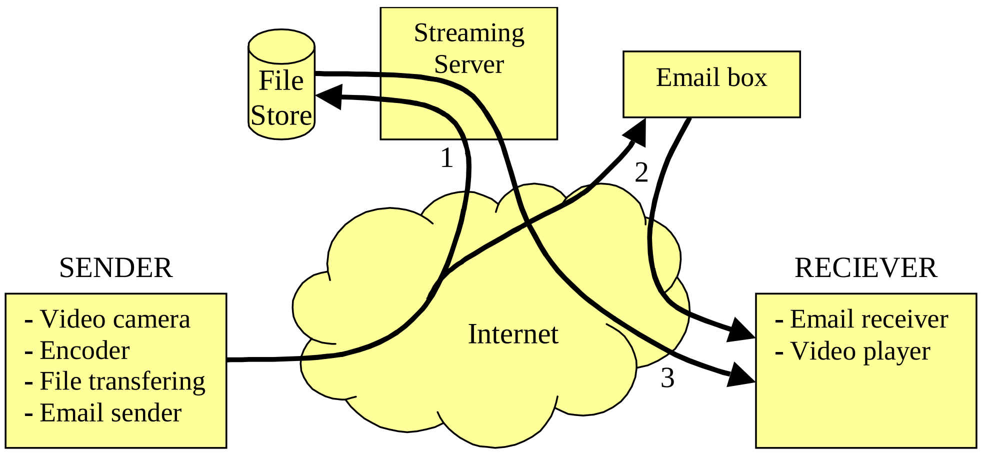


Figure 3.8: mVideo Messaging sending model

The sender agent is equipped with a video camera. The video signal is compressed and encoded in PVAuthor, which creates a locally stored MPEG-4 file. This file is first transferred to the server where it is stored. When the file is completely uploaded, an email is sent to the recipient. This means that an mVideo Message consists of a link to a video clip file and an email as shown in figure 3.9.

[53] The new possibilities next generation's network makes possible are described in section 2.2.1.

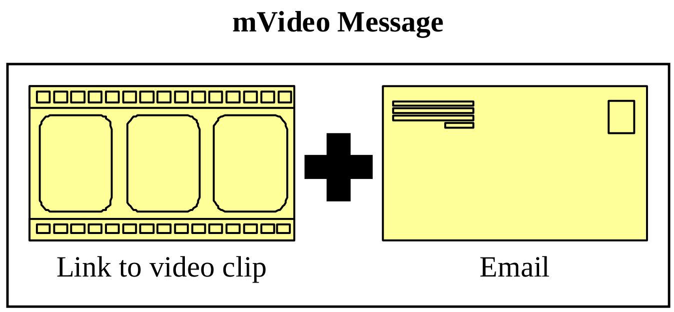


Figure 3.9 mVideo Messages consist of an email with a link to a video-clip file

The recipient agent is continuously checking for new "mVideo-email". When an email has arrived, the mVideo-application makes use of information within the email and displays a notification for the user. The notification tells that a new mVideo message has arrived and asks the user whether he/she wants to view it. If the user decides to view the message, PVPlayer is launched with a link to the video file at the server. This starts the streaming video session and the message is displayed on the mobile device.

3.3.3 Camera emplacement discussion

This section involves a discussion of where on the sender agent the video camera should be connected and how the two agents are to communicate. This discussion fulfills the high-level design for the sender agent. This section is not necessary to read for the understanding of how the prototype works but is needed to motivate the high-level design.

The sender agent will be based on a handheld device (an iPAQ) and a Microsoft Windows 2000 equipped laptop computer (throughout the rest of the report the term "laptop" will be used instead of "Microsoft Windows 2000 equipped laptop computer"). The receiver agent will only be based on a handheld device (an iPAQ).

An mVideo message consists of a video clip stored in the encoded and compressed file format called MPEG-4, and an email. A video camera, a laptop, and an iPAQ are the hardware needed to create and send an mVideo message. The encoder software called PVAuthor that is running on the laptop has the possibility to take a video signal as input. This could be done by simply installing and connecting the video camera to the laptop and choosing the camera as source in PVAuthor. PVAuthor also has the possibility to take an AVI-file as input. This creates four options of where to have the camera and where to create the MPEG-4 file.

1. Having the camera connected to the laptop and encode the video signal to an MPEG-4 file directly.

2. Having the camera connected to the laptop and first create an uncompressed avi-file containing the video and then encode it to MPEG-4 format.
3. Having the camera connected to the iPAQ and transfer an uncompressed avi-file directly to the laptop and then encode it to MPEG-4 format.
4. Having the camera connected to the iPAQ and store an uncompressed avi-file locally and then transfer the file to the laptop.

We made the design decision to use the first alternative. The second and fourth option implies that the mVideo-application in some how needs to create an avi-file. Since a 20 seconds video clip in the avi-format is about 50 Megabyte, there is no possibility, due to the memory limitations of the iPAQ, to first store the file locally. This makes the fourth alternative impossible and it is hard to find advantages of first storing the file locally at the laptop and then start encoding when there is a possibility of encoding directly.

Our research shows that option number three can not be achieved easily. The main reason is that in order to connect the video camera to the iPAQ, we have to use an expansion pack system that enables users to add industry standard PC and Compact Flash Cards to the iPAQ. With the expansion pack it should be possible to plug a video-card into the iPAQ and then connect the video camera to this video card. In figure 3.10 an iPAQ with expansion pack is shown.

Figure 3.10: An iPAQ with expansion pack [8]

As we concluded above, the iPAQ also needs to be able to communicate with the laptop, since it has to send the video signal to the encoder software. The iPAQ has an IR-port that in theory could be used for this, however this is not very practical because of the very limited range. An alternative that we found very attractive was to use a Radio LAN PC-card for this communication. But since only a "single-slot" expansion pack was available at this time (April 2001) there was no possibility of connecting the video camera to the iPAQ and still being able to handle the communication.

These are the motives to use option number one - to have the camera connected to the laptop and encode the video signal to an MPEG-4 file directly. This means that the iPAQ will be used as a remote control for the recording part. The main purpose of the iPAQ within the application is to simulate a future 3G-telephone.

The receiver agent is not dependent on the laptop but only to an Internet connection. There are many possibilities to connect the iPAQ to the Internet. The two different channels we tried with good results are to use a Nokia Cardphone that provides a GSM connection, and to use a radio LAN network card that provides a 2 MB connection[54]. In figure 3.11 a Nokia Cardphone and an Orinoco W- LAN PC Card is shown.

Figure 3.11: A Nokia Card Phone and a W-LAN PC Card

3.3.4 Sending mVideo Message

The discussion above ended the high-level design for the prototype. In this section the application part of the sender agent is described, but before we go into the details, an overview of how the sender agent works is given.

The iPAQ works as a wireless remote control for the laptop. A video camera is connected to the laptop and its signal works as input to compression and encoding software running on the laptop. The remote control should provide the functionality to record a video clip and then send an mVideo message to a recipient. The main task of the laptop is to record video material, encode it to MPEG-4 format, transfer the encoded file to a streaming video server on the Internet and finally send an email to a recipient with information on how to reach the video file (see figure 3.12).

[54] The radio LAN speed depends upon how the radio network is connected to Internet.

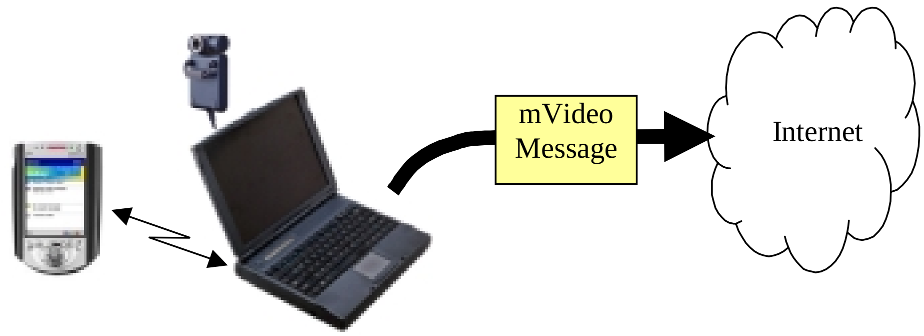


Figure 3.12: The mVideo Messaging Sender agent

To be able to develop a prototype with the behavior described above, two programs are to be developed - one program for the laptop and one for the iPAQ. The program for the laptop is called MMCreator and the program for the iPAQ is called MMClient.

The creation and sending of an mVideo message consist of three user scenarios: Start recording, Stop recording, and Send message.

1. **Start recording a video clip**. The user chooses to start recording an mVideo message on the iPAQ. MMClient sends requests to MMCreator, which in turn tells the camera to start recording and PVAuthor to start encoding.
2. **Stop recording a video clip**. When the user is satisfied with recording, he/she chooses to turn the recording off. MMClient sends requests to MMCreator to end the recording session and stop the encoding. The encoded video clip is stored as an MPEG-4 file on the laptop. MMCreator sends over the size of the file to MMClient which uses the information to calculate a price for the coming transmission over Internet.
3. **Sending an mVideo message**. The user fills in the information needed for sending and chooses to send the mVideo message. MMClient sends over the information to MMCreator. MMCreator renames the video file to a unique name, uploads the file to the MMS Server and then builds an email upon the information given by the user and some data about the video file. The email is sent to a recipient via an email server.

Design discussion for the sender part of MMClient. We had several ideas on how to realize MMClient. One was to work in HTML[55] because of its advantages when it comes to building graphical interfaces. Working in HTML would also have made MMClient, in a way, platform independent. The idea would have been to implement some static HTML-pages on the iPAQ and some pages on a web-server located on the laptop. The HTML-

[55] HyperText Markup Language

pages on the web-server would have communicated with MMCreator via Java-servlets. This would, as we will see in the next section 3.3.5, made the new-message-notification hard to implement. The possibility we concluded was to build that part in a separate programming language. Because of this we decided to build the MMClient in a programming language that allowed us to write both the sender agent and the receiver agent in the same language. This opened up the possibility of integrating these two agents in the same application.

The programming languages we chose between were embedded Visual Basic and embedded Visual C++. Embedded Visual Basic was chosen instead of embedded Visual C++ because it would be easier to work with and quicker to develop due to our previous experience.

MMClient and MMCreator exchange information using socket-communication. Sockets are a standard way to communicate over an IP-network.

Design overview for the sender part of MMClient. We had a few ideas when it came to how to divide the logic in MMClient. What we wanted to do was to use a three-layer model with a presentation layer, a business layer, and a data layer. In this model, the presentation layer is the user interface, the business layer incorporates the business logic (the part of an application program that performs the required data processing of the business) and the data layer takes care of all data. It proved to be hard and time-consuming to use this model in Visual Basic, so we had to simplify this idea. Instead we are using a two-layer model with a presentation layer and an “engine-layer”. The engine-layer deals with the communication with MMCreator but as we will see in next section, when the receiver agent is to be built, this layer will also take care of communication with a pop-server and a database. In figure 3.13 the two-layer model used for the sender part of the agent is shown.

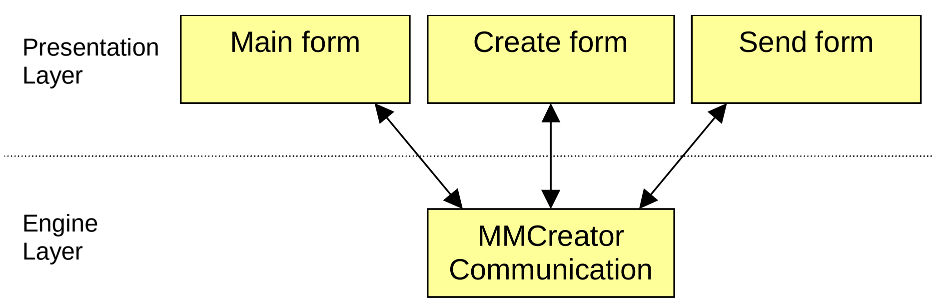


Figure 3.13: Modules and mutual dependencies for sender part of MMClient

The Main form is the start page of the application. From here, a user has access to all the features of MMClient. The Create form is used when creating new mVideo messages. The form lets the user start and stop recording by calling methods in the MMCreator Communication module. The Send form allows the user to send the mVideo message in the Create form. In this form additional information such as receiver is added. By calling methods in MMCreator Communication the mVideo message is sent.

The graphical interface for sender part of MMClient. When the GUI[56] was designed, the goal was to design it to be representative of a future cell phone. We tried to minimize the use of the iPAQ pen stylus and make the GUI more "thumb-clickable". In order to achieve this we used big buttons and menus and tried to avoid text input as much as possible.

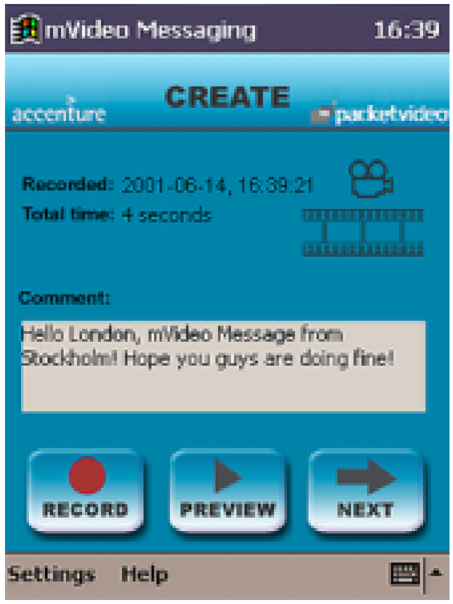
We also wanted to reduce the user learning time and ensure that the user quickly felt "at home" with the application. We wanted the system to be a "walk-up-and-use" system, i.e. anyone should be able to use the system at a first try. Due to this demand we have looked to existing and popular services like SMS and email when designing the application.

The start page has been stripped down in order to set the focus on what we believe is essential. Functions for help and settings have been moved down to a lower menu-bar.

Main form

From the main form, or start page, a user can choose to create and send an mVideo message, look at his/her inbox for received messages or at messages sent.

[56] Graphical User Interface



Create form

The creation and sending of messages is divided into two steps. In the first a user makes the actual recording and fills in an optional comment. Some information about the date, time and length of the video clip is shown. After recording, the user can view a preview and/or press the Next-button to proceed.

Send form

After pressing Next at the Create Message page, the user comes to Send Message. A receiver is selected from the contact-list[57]. Then the user chooses in what mode he/she wants to send the message. Standard means that the message may take a while longer to send (e.g. is sent during low-cost hours) and Express means that the message is sent instantly. The price for the different modes is based on the size for the encoded video file[58]. The user sends the message by pressing the Send-button.

With these screenshots we are leaving the sender part of MMClient and continue to describe the Java program called MMCreator.

[57] This list is directly connected to the Contacts function in the iPAQ

[58] After the video clip is recorded, MMCreator sends over the size of the file to MMClient. MMClient uses this information to calculate price for Standard and Express mode.

Design discussion for MMCreator. The MMCreator program was developed in Java due to the advantages of the language when it comes to platform independence and the previous knowledge of the programmers. The encoded video file is transferred from MMCreator to the MMS Server using FTP[59]. It is the standard procedure to transfer files over the Internet and it is easy to start ftp-services on the MMS Server. This means that no own-developed program needs to be developed for the MMS Server.

Design overview for MMCreator. We decided to divide the MMCreator into four different modules according to figure 3.14.

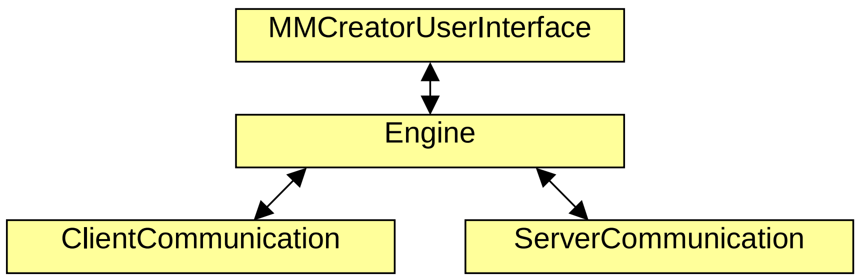


Figure 3.14: Modules and mutual dependencies for MMCreator

MMCreatorUserInterface provides some basic administrator-functionality for MMCreator. The Engine module is the heart of MMCreator. This module starts up the system and here the MPEG-4 files are also created. The ClientCommunication module handles all communication with the user-application MMClient. The ServerCommunication handles uploading of the MPEG-4 files to the MMS Server and sending of email through an email server.

All necessary configurations for the MMCreator, such as IP-addresses to MMS Server and email server, are done in the MMClient settings and are sent over to MMCreator when user presses the send button.

3.3.5 Receiving mVideo Message

In this section, the receiver part of the agent is described. First an overview of how the receiver agent works is given which is followed by a more detailed description.

When the receiver gets an mVideo message, the first thing that happens is that an email is delivered to the receiver agent. This email contains information about the sender, comments, references to the video file etc. The receiver agent uses this information to display a notification for the user.

[59] File Transfer Protocol, a protocol used to transfer files over a TCP/IP network.

This notification shows information about the message and asks the user whether he/she would like to view the video message. If the user answers yes, a video player is launched with a reference to the video file and the video message is automatically started. This course of events is shown in figure 3.15. All messages are stored in an inbox on the iPAQ so if the user does not want to see the message immediately it can be found in the inbox later. Note that the video file itself is never downloaded or stored on the handheld device but only information about how to reach it on the MMS Server.

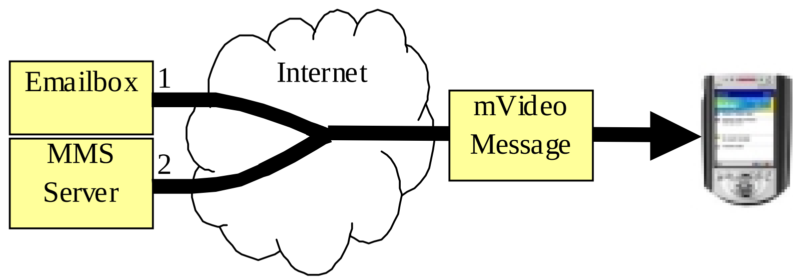


Figure 3.15: Course of events to receive an mVideo message

Design discussion for receiver part of MMClient. In the previous section we concluded that we had several ideas on how to realize MMClient. As been concluded above, we want to integrate the receiver part of the agent with the sender part. We think it would be unnatural to be able to send but not receive messages within the same application.

One early idea was to work in HTML because it's very easy to build a graphical interface in it. Working in HTML would have made MMClient, in a way, platform independent. But in PocketExplorer, the standard web-browser on the iPAQ, you are only allowed to have one browser-window open simultaneously. To be able to continuously check for new email with the program this would occupy the web-browser all the time. This means that a user could not utilize the web-browser for other purposes and the mVideo application could not run as a background service. Checking for new email in an HTML-solution would therefore present major disadvantages on the iPAQ.

An alternative is to have a separate solution for the checking part, for example to write a small non-HTML program that could do this as a background service. But this kind of solution would fragment the application in a way we did not want to. Instead of building a separate notification part, we decided to build the whole MMClient independent of the web-browser. This was done in embedded Visual Basic.

To check for new email MMClient needs to connect to an email-server. This is done with sockets. Sockets are a standard way to communicate over an IP-network.

Design overview for MMClient. In section 3.3.4 we took the decision to use a two-layer model with a presentation layer and an “engine-layer”. To be able to add the extra functionality, necessary for the receiving part, the design model is extended. In figure 3.16 the two-layer model used is shown, the parts dealing with the creation and sending are highlighted using dotted edges.

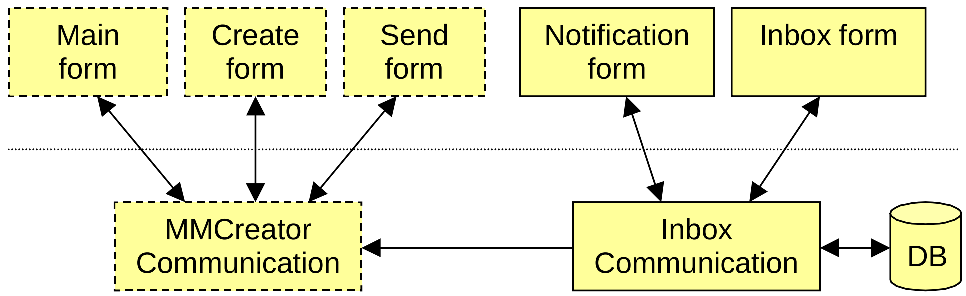


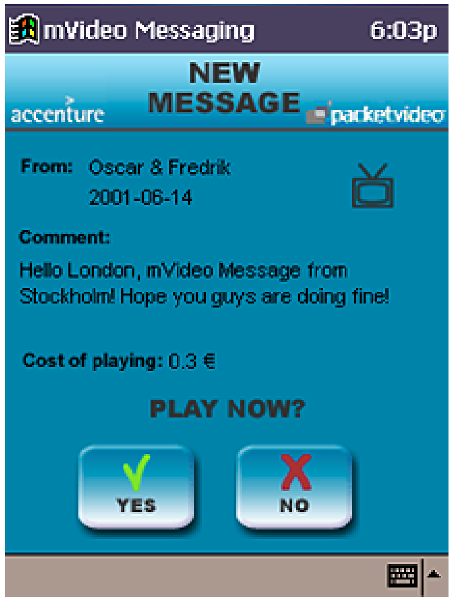
Figure 3.16: Modules and mutual dependencies for MMClient

The Notification form is displayed for the user when a new mVideo Message arrives. The process that is continuously checking for new email is running in the Inbox Communication module and as soon as a new mVideo email arrives, this module activates the notification form.

The Inbox form presents a list of old and/or new messages to the user. All data concerning the inbox (e.g. old messages) and all configuration settings for the application are stored in the database. Some settings are information that is required for the MMCreator Communication module (e.g. IP-address for MMCreator and the MMS Server) and therefore the arrow from the Inbox module to the MMCreator module.

The graphical interface for receiver part of MMClient. The same ideas as for the sender part have inspired the graphical user interface for this part.

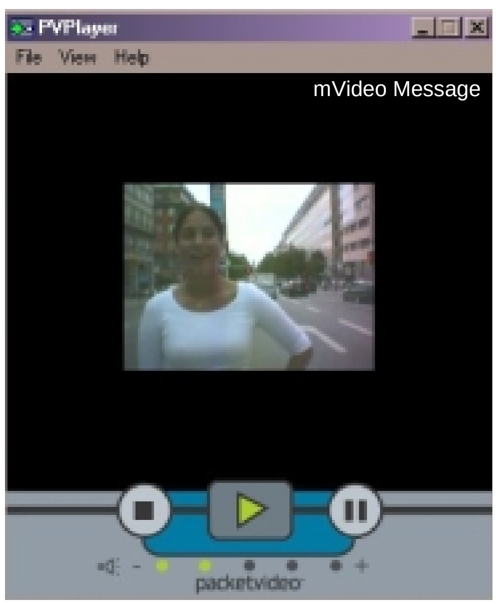
Also here, we have looked to existing and popular services like SMS and email when designing the application. Examples here are the use of the term’s "inbox" and "forward" (compare with email) and that a user gets a notification asking if the message is to be shown. Furthermore, it might be worth noting that it was clear from the beginning that it was not possible to integrate PVPlayer in the application.



Notification

Upon receiving an mVideo message, a notification page is shown. This page displays information about the sender, date of sending, the comments and the cost of viewing. If the user chooses to view the message by pressing the Yes-button, PVPlayer is started and the video is streamed from a server to the iPAQ.

e



The Player

When the video part of the message is shown a special player is launched. The user can play, pause and stop the video session. The volume can also be adjusted.

Inbox

The inbox shows received messages. An icon shows what kind of message the row is representing –video, sound, or only text (ordinary SMS). When the user marks a message, the comment for the actual message is shown. The messages can be deleted, forwarded and played. The play function launches the player.

These pictures end the section about the receiver part of the agent. In the next section possible improvements for the prototype are presented.

3.3.6 Improvements and limitations

Due to the purpose of the prototype and the limited time available (two months) for it's development, several simplifications were done. In this section, some more of the demands that a full-scale working product would need to support are presented.

- **Integrate the camera and encoder in the device.** The fact that we could not connect the camera to the iPAQ and do the encoding on it makes the look of the sender part very different from a next generation cell phone. The use of a laptop for this decreases the mobility considerable. We think a good improvement would be to integrate these aspects into the handheld device.
- **Increase the number of data carriers.** The prototype is limited to use email to reach the receiver with message data. A good improvement for a real service would be to let the MMS Server have the possibility to choose whether email or SMS would be the best method of informing the receiver. This would increase the chances of reaching the receiver, for an example, during a system failure at one of the carriers.
- **Increase the number of message types.** The prototype is limited to sending video messages. A good improvement would be to make it possible to send text or sound messages as well. This would make use of the different icons showing message type in the inbox and in a real service the MMS should provide the SMS as well.
- **Make it possible to send a message to a group of receivers.** The prototype is limited to only one receiver. We think a good function could

be to be able to send a group message to many receivers. It would be very easy to implement this functionality to the prototype, but due to the lack of time this has not been done.

- **Increase the security on the MMS Server.** The MMS Server we use has not been built with any security in mind. All messages are uploaded to a public folder on the server, which means that all users are able to access all messages that are sent through the system. This simplification is not acceptable for a real service. Security issues were not to be covered within this prototype.

3.4 Conclusions

Chapter 3 deals with the second issue for the master thesis – to build a prototype application for a multimedia message service, using a suitable streaming video platform. To set extra focus on the video ingredient, the prototype is called mVideo Messaging.

The heart of the mVideo Messaging is a streaming video platform. The two different streaming video platforms evaluated and compared are the PacketVideo PVPlatform and the Emblaze Systems Wireless Media Platform. PacketVideo and Emblaze are both members of the Wireless Multimedia Forum and their platforms are designed according to the WMF end-to-end system model. Therefore the platforms look very much the same, but with some essential differences. The PacketVideo platform includes software that handles variations in available network bandwidth by allowing the video player to continuously report network conditions to the streaming server and in addition the server will adjust streaming rates to accommodate changes. The platform used within the prototype is the PacketVideo platform.

The service and the prototype developed should be representative for a service on a future 3G-cell phone. Since tomorrow’s cell phones are not available at the moment for the thesis, a Compaq iPAQ is used instead.

The characteristics of the streaming video platform makes it necessary to use a laptop when creating a multimedia message. Therefore the prototype is divided into one sender agent consisting of an iPAQ, a laptop and a video camera, and one receiver agent consisting of an iPAQ. In a real situation we think the camera and the encoder should be integrated in the cell phone.

The multimedia message consists of an email with a link to an MPEG-4 file that resides on an MMS server. This means that the video file itself is not transferred to the receiver agent but instead it is streamed when played.

The multimedia message service proposed in chapter 2 is practicable. In this chapter it is also shown that the service can be built using the technology of today. A Nokia Card phone that was plugged into the iPAQ provided a GSM channel at about 30 kbps to stream through. This proved to work well.

4 Streaming video theory

This chapter gives a theoretical background that is needed to deal with the third issue for the thesis – to present problems and technical solutions that are associated with video streaming. In chapter 5 the streaming video platform will be tested in an environment that emulates different kind of network situations. The tests are partly done in order to see how theories given in this chapter works in practice.

In 1948 Claude Shannon introduced the main concepts of what has become the field of Information Theory [50]. His communication system model is shown in figure 4.1. This model is used in order to divide this chapter into two separate parts, where the first part describes how the network that the streaming video is sent through works. This could be said corresponds to the right part of the communication model. In the second part theories for image coding and data compression are presented.

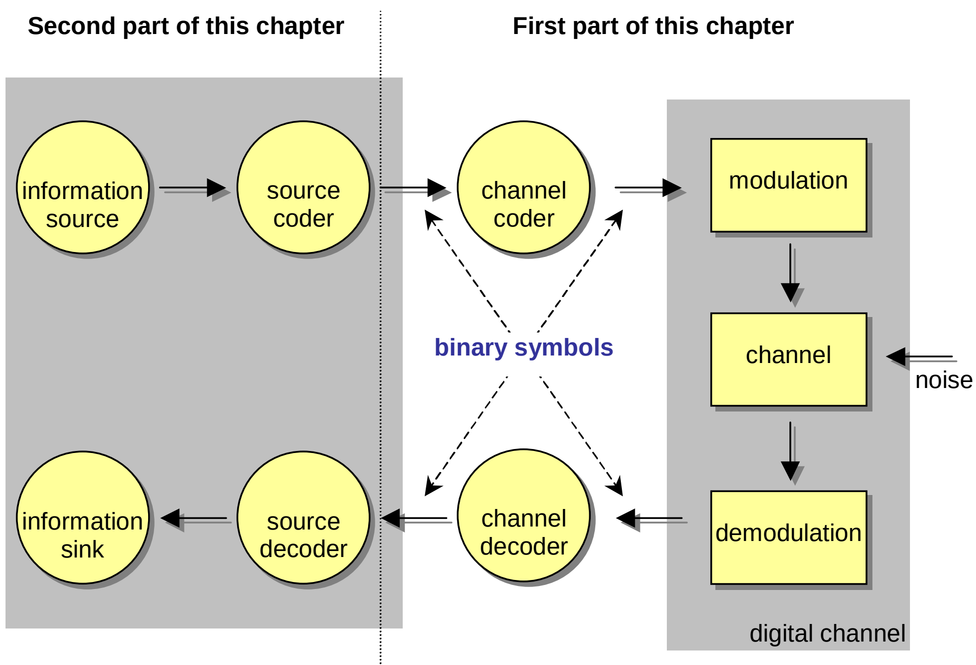


Figure 4.1: Communication system model [50]

4.1 Streaming in networks

Figure 4.2 shows the mVideo Messaging Service architecture that was given in section 3.3.2. The video stream goes from the MMS Server through Internet, into a mobile network and ends up in a video player located in the mVideo Messaging Agent. In this section we first take a look at channels for the video stream – Internet and Mobile Networks. After concluded the limitations and problems for these channels we describe how the streaming server and the video player acts to provide good video quality under the given conditions.

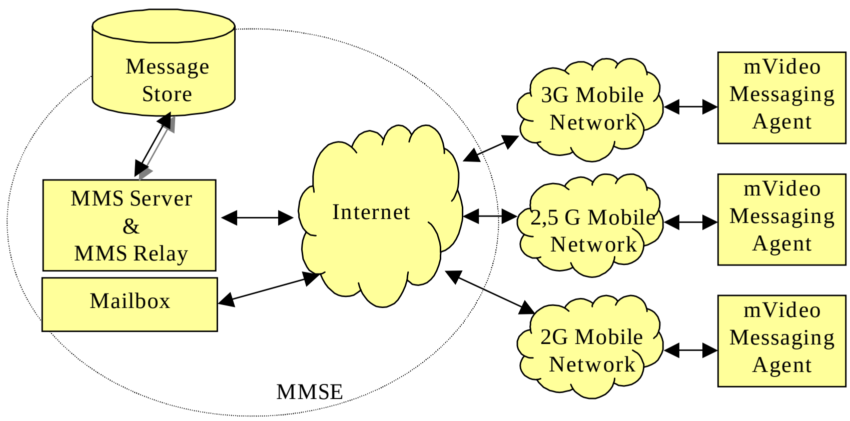


Figure 4.2: The mVideo Messaging Service architecture

4.1.1 Internet basics

In this section we examine the basic structure for communication over the Internet. An example of how a simple text message could be sent between to computers over the Internet will give us a basic understanding before going into the details in next section.

Because the Internet is a global network of computers each computer connected to the Internet must have a unique address. Internet addresses are in the form nnn.nnn.nnn.nnn where nnn must be a number from 0 - 255. This address is known as an IP address. Figure 4.3 illustrates the situation (simplified) for the mVideo prototype.[41]

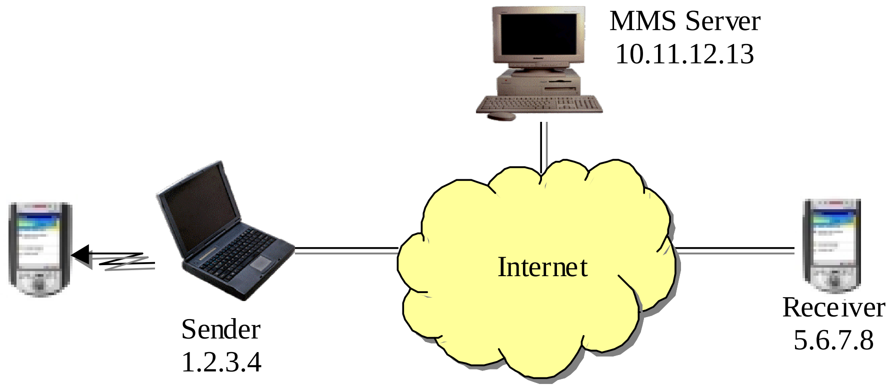


Figure 4.3: The Internet addressing situation for the mVideo prototype

If a computer is connected to the Internet through an Internet Service Provider (ISP), it is usually assigned a temporary IP address for the duration of the dial-in session. This is the case for the mVideo receiver when we used the Nokia Cardphone. [41]

So the computers in figure 4.3 are connected to the Internet and have a unique address. But how do they communicate with each other? An example should serve here: Let's say the sender computer with IP address 1.2.3.4 want to send a message to the MMS Server with IP address 10.11.12.13. The message to send is "Hello computer 10.11.12.13!". Obviously, the message must be transmitted over whatever kind of wire connects the sender computer to the Internet. Therefore the message must be translated from alphabetic text into electronic signals, transmitted over the Internet, and then translated back into alphabetic text. This is accomplished through the use of a protocol stack. Every computer needs one to communicate on the Internet and it is usually built into the computer's operating system. The protocol stack used on the Internet is called the TCP/IP protocol stack because of the two major communication protocols used. The look of the TCP/IP stack is shown in figure 4.4. [41]

Protocol Layer	**Comments**
Application Protocol Layer	Protocols specific to applications such as WWW, e-mail, FTP, etc.
Transport Layer	Directs packets to a specific application on a computer using a port number.
Internet Layer	IP directs packets to a specific computer using an IP address.
Hardware Layer	Converts binary packet data to network signals and back. (E.g. Ethernet network card, modem for phone lines, etc)

Figure 4.4: The TCP/IP stack [41]

To follow the path the message "Hello computer 10.11.12.13" took from the sender computer to the MMS server it would happen something like this:

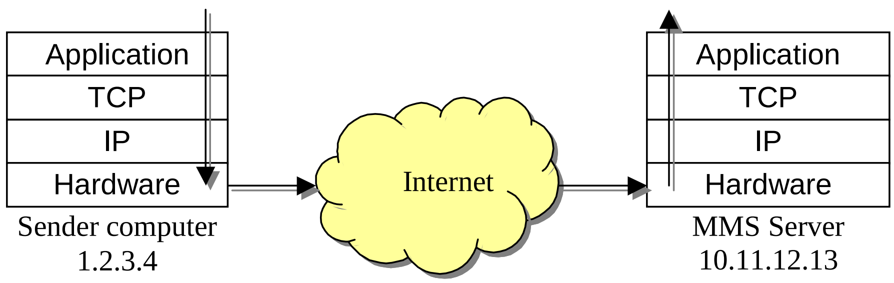


Figure 4.5: The path for a message through the TCP/IP stack [41]

The message starts at the top of the protocol stack on the sender computer. If the message to be sent is long, each stack layer that the message passes through may break the message up into smaller chunks of data. This is because data sent over the Internet are sent in manageable chunks, called packets. [41]

The packets would go through the Application Layer and continue to the TCP layer. Each packet is assigned a port number, this is because many programs may be using the TCP/IP stack and sending messages. We need to know which program on the destination computer needs to receive the message because it will be listening on a specific port. After going through the TCP layer, the packets proceed to the IP layer. This is where each packet receives its destination address, 10.11.12.13. [41]

Now that the message packets have a port number and an IP address, they are ready to be sent over the Internet. The hardware layer takes care of turning our packets containing the alphabetic text of our message into electronic signals and transmitting them over the wire. [41]

The packet are sent to a router that examines the destination address in each packet and determines where to send it. Often, the packets next stop is another router. Eventually, the packets reach computer 10.11.12.13. Here, the packets start at the bottom of the destination computers TCP/IP stack and work upwards. As the packets go upwards through the stack, all routing data that the sending computers stack added (such as IP address and port number) is stripped from the packets. When the data reaches the top of the stack, the packets have been re-assembled into their original form, "Hello computer 10.11.12.13!" [41]

The Internet is a shared medium and uses a best effort delivery mechanism to deliver content. There is no dedicated path between source and the target. The Internet Protocol breaks content into packets and these packets are routed independently. Limited bandwidth, noise, packet loss, retransmission and out of order packet delivery are all problems when sending packets over Internet. Traditional Internet traffic deals with these problems high up in the protocol stack at its leisure. But for Live or on-demand streaming it is critical to get the packet in time. Streaming is very sensitive to the variation in delay that is common in Internet-traffic. [9]

To see what opportunities the TCP/IP model gives to deal with these kinds of problems we take a deeper look at the TCP/IP model in the next section.

4.1.2 TCP/IP reference model

As mentioned in previous section, the Internet reference model contains four layers: the hardware layer, the Internet layer, the transport layer, and the application layer. In the following sections, we describe the function of each layer in more detail, starting with the hardware layer and working our way up to the application layer.

Hardware layer. The hardware layer is the lowest layer in the TCP/IP reference model. This layer contains the protocols that the computer uses to deliver data to the other computers and devices that are attached to the network. The protocols at this layer define how to use the network to transmit a frame, which is the data unit passed across the physical connection. The protocols here also exchange data between the computer and the physical network. They deliver data between two devices on the same network. [6]

The TCP/IP reference model only point out that the host has to connect to the network using some protocol so it can send IP packets over it. This protocol is not defined and varies from host to host and network to network. [52]

Internet Layer. The layer above the hardware layer is called the Internet layer. This layer is responsible to permit hosts to inject packets into any network and have them travel independently to the destination. They may even arrive in a different order than they were sent, in which case it is the job of higher layers to rearrange them.

The Internet layer defines an official packet format and protocol called IP (Internet Protocol). The job of the Internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly the major issue here, as is avoiding congestion. [52]

Transport Layer. The layer above the Internet layer is usually called the transport layer. It is designed to allow peer entities on the source and destination hosts to carry on a conversation. Two end-to-end protocols have been defined here. The first one, TCP (Transmission Control Protocol) is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error to any other machine on the Internet.

The second protocol in this layer, UDP (User Datagram Protocol), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own. UDP is used in place of TCP when reliable delivery is not required. For example, UDP is used for realtime audio and video traffic where packetloss are simply ignored, because there is no time to retransmit. [52]

Application Layer. On top of the transport layer is the application layer. It provides the services that user applications use to communicate over the network. This is also the layer where processes for user-access to the network reside. The application layer contains all the higher-level protocols, such as hypertext transfer (HTTP), file transfer (FTP), and electronic mail (SMTP). [52]

The application layer also manages the sessions (connections) between cooperating applications. In the TCP/IP protocol hierarchy, sessions are not identifiable as a separate layer, and the transport layer performs these functions. Instead of using the term "session," TCP/IP uses the terms "socket" and "port" to describe the path over which cooperating applications communicate. [6]

The relation of IP, TCP, and UDP is shown in figure 4.6. Since the model was developed, IP has been implemented on many other networks. [52]

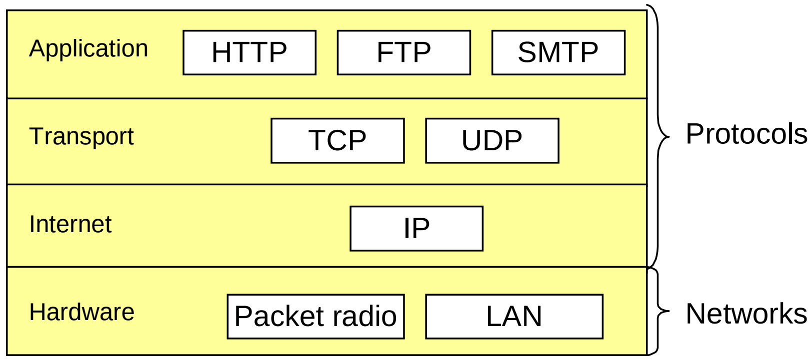


Figure 4.6: Examples of streaming protocols and networks in the TCP/IP model [52]

The TCP/IP model, as described above, is a "best-effort" service for all the datagrams it carries. Best effort does not make any promises about end-to-end delay for an individual packet, or variation of packet delay within a packet stream. Neither TCP nor UDP can make any guarantees of delays to applications that utilize these protocols. This is not a problem for email or web traffic, but for multimedia applications including for example streaming this could be a major problem. [40] In the next section 4.1.3 we take a deeper look at these Internet related problems.

4.1.3 Internet problems

There are, in principal, three major problems for multimedia applications when sending data over Internet: Packet loss, end-to-end delay, and delay jitter. In this section we explain what these problems depend upon and how to deal with them.

Packet loss. As an IP datagram wanders through a network, it passes through buffers in the routers in order to access outbound links. It is possible that one or more of these buffers in the route from sender and receiver is full. In this case, the IP datagram is discarded and it will never arrive to its destination. This problem is solved in TCP (it retransmits lost packets) but retransmission mechanisms are not often used in e.g. real-time audio applications because they increase the end-to-end delay.

Loss Recovery Schemes could be used to overcome packet loss. A packet is lost if it never arrives at the receiver, or if it arrives after its scheduled playout time. Real time applications utilize loss anticipation schemes such as forward error correction and interleaving.

Forward error correction makes use of redundant information that has been added, to construct approximations or exact versions of lost packets. Before transmission, the data is processed through an algorithm that adds extra bits for error correction. If the transmitted message is received in error, the correction bits are used to repair it. [40]

Alternative to sending redundant information is sending interleaved chunks. As an example: If units are 5 ms in length, and chunks are 20 ms (4 units per chunk), then the first chunk could contain units 1,5,9,13, the second chunk could contain 2,6,10,14, the third chunk 3,7,11,15 and the fourth chunk 4,8,12,16. Loss of a single packet, for example chunk number two, will result in four small gaps in the re-constructed stream instead of one large.

The re-constructed interleaved stream would look like this: 1,_,3,4,5,_,7,8,9,_,11,12,13,_,15,16.

The re-constructed non-interleaved stream would look like this: 1,2,3,4,_,_,_,_,9,10,11,12,13,14,15,16.

The interleaved alternative has a significant improvement in perceived quality since a small gap is not so easily noticeable as a large gap. [40]

End-to-end delay. The end-to-end delay is the accumulation of transmission processing and queuing delays in routers, propagation delays, and end-system processing delays along a path from source to destination. [50]

For applications such as Internet Phone, delays below 150 ms are not perceivable, delays between 150 ms - 400 ms can be tolerated, but anything beyond 400 ms is not acceptable. Packet delayed more than 400 ms are generally ignored by the receiver, but this is a threshold limit that the receiver sets. [40]

Delay jitter. A crucial component of end-to-end delay is the random queuing delays in the routers. Because of these varying delays within the network, the time from when a packet is generated at the source until it is received at the receiver at the end point can differ from packet to packet. It is this that is called jitter. [50]

If the receiver ignores the jitter and just plays out packets as soon as they arrive, the result will not be very good. Jitter can be handled by using sequence numbers, timestamps, and playout delays.

Sequence numbers means that the sender places a number on each of the packets and increments it for each packet. Timestamps means that the sender stamps each packet with the time at which the packet was sent. Fixed playout delay means that the receiver attempts to play out each chunk exactly $q$ms after the chunk has been generated, provided it has been received. Hence, if a chunk is timestamped at t, playout is at $t+q$. Packets arriving after their scheduled playout times are discarded and considered to be lost. By making the initial playout delay large, most packets will make their deadlines. [40]

4.1.4 IETF streaming protocols

What could be done with the Internet to better accommodate to multimedia traffic? Adding more bandwidth to the delivery channels is of course one way. Another option is to use new protocols that are more suitable for streaming over the Internet. The IETF (Internet Engineering Task Force) is a voluntary standards body dedicated to making recommendations for how to communicate information over IP networks. The IETF has a number of activities related to the delivery of multimedia information in packed-based environment. For the delivery of video streams with synchronized audio from a server to a terminal IETF recommends the use of methods based on RTP and RTSP. [34] In this section the TCP/IP reference model, described previously, is modified according to IETF's recommendations in order to better deliver one-way multimedia data over IP networks.

Real Time Streaming Protocol (RTSP). RTSP is a session-oriented protocol that is transported over TCP between server and client. The purpose of RTSP is to provide a language for communicating standard video-on-demand requests. [34] RTSP establishes and controls either a single or several time-synchronized streams of continuous media such as audio and video. It does not deliver the continuous media stream itself, although interleaving of the media stream with the control stream is possible. In other words, RTSP acts as a "network remote control" for multimedia servers. Such control actions include pause/resume, repositioning of playback, fast forward and rewind. [28]

There is no notion of an RTSP connection; instead a server maintains a session labeled by an identifier. During a RTSP session, an RTSP client may open and close many reliable transport connections to the server to issue RTSP requests. Alternatively, it may use a connectionless transport protocol such as UDP. [28] Consequently, RTSP does not define how audio and video are encapsulated in packets for transmission; instead this is defined via RTP [40].

Real Time Transport Protocol (RTP). The idea behind RTP is that certain data needs to be delivered from a server to a client in a real time manner. RTP is an application layer component that utilizes UDP as transport mechanism and an RTP packet consists of sequence numbers, timestamps, and payload. RTP enables a client application to monitor the loss of packets, and to "re-order" those packets that arrive out of order at the client. But RTP does not address resource reservation and does not guarantee quality-of-service for real-time services. [4]

Real Time Control Protocol (RTCP). RTCP is a sub-component to RTP that is used to control performance information between server and client. This information could be such as the percentage of RTP packet loss during a video session, which is crucial to managing the quality and throughput of the video data from the server.[34] Both RTCP and RTP are designed to be independent of the underlying transport and network layers. [4]

The network protocol stack for delivering one-way multimedia data is shown in figure 4.7.

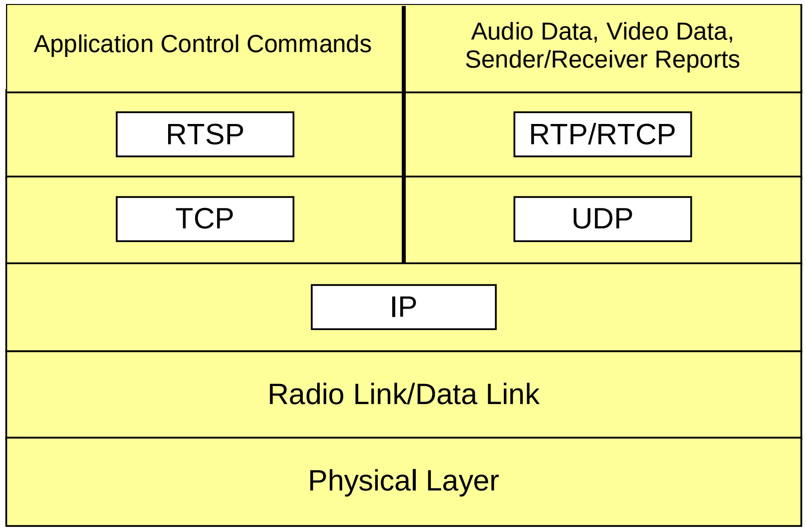


Figure 4.7: Network protocol stack for delivering one-way multimedia data [34]

In a wired network bit errors are very rare and network congestion is the most likely source of packet loss. But the way TCP provides reliable end-to-end service in the Internet can result in problems when TCP/IP is run over wireless links. Error recovery in the Internet is typically handled at the transport layer by TCP, with IP providing a basic unreliable service at the Internet layer. This allow applications that not require reliable service to use another end-to-end protocol such as the UDP. However, research indicates that link layer error recovery schemes over wireless Internet links can improve the performance of higher layer end-to-end protocols.

The link layer approach to error recovery is both potentially faster than end-to-end recovery, and adaptable to the wireless link characteristics. The approach is to handle wireless link errors at the link layer by implementing a protocol that hides errors from the higher layers. [23] The next section gives a brief description of the error recovery characteristics for the link layer protocol in the 3G mobile networks.

4.1.5 TCP/IP over 3G mobile networks

Much development and deployment activity has focused on UMTS and IMT-2000, also referred to as 3G wireless networks. Since a primary motivation for this is data communication and Internet access, TCP performance is a key issue. As we concluded in previous section, the link layer characteristic affects TCP performance over the link. [29]

The 3G wireless networks are based on W-CDMA[60] that uses the RLC[61] protocol. RLC uses sequence numbers to detect a gap between received frames. When detecting such a gap, RLC can issue a status report and invoke retransmission. This means that RLC preserves order of packet delivery. [29]

The W-CDMA also makes use of forward error correction and interleaving (described in section 4.1.3). In general, W-CDMA can provide a packet service with a negligible small probability of undetected errors, and a low level of packet loss for the upper layer traffic. But the retransmission introduces latency and jitter that results in relatively large Bandwidth-Delay Product (BDP) of the 3G wireless networks. [29]

From the perspective of transport layer, the underlying W-CDMA network can be viewed as a network with a relatively large BDP and jitter. The loss rate of IP packets is low due to the retransmission, but the recovery appears as jitter to the higher layers. [29]

This section about packet delivery over 3G ends the survey of the conditions for streaming over Internet and 3G mobile networks. In the previous section we have seen how the conditions for packet delivery changes when sending them over a wireless channel. This has implied that the error recovery mechanism in the TCP/IP reference model have been moved from the transport layer down to the link layer. We have also introduced protocol that is suitable for streaming over Internet, making it possible to e.g. pause/resume, fast-forward, and rewind the stream.

4.1.6 Streaming video techniques

In this section we describe how the techniques given in the previous sections are used in multimedia streaming platforms.

Streaming media consists of streaming audio and streaming video. The term “streaming” implies a one-way transmission to the viewer, in which both the client and server software cooperate for uninterrupted motion. The client side buffers a few seconds of video or audio data before it starts sending it to the screen, which compensates for momentary delays in packet delivery. Streaming media solutions are usually based on an architecture comprised of a source, an encoder, a streaming server, and a player. To recall the architecture used within the mVideo prototype the PacketVideo system model is shown in figure 4.8. The content is the source, PVAuthor is the name of the encoder, PVServer is the streaming server, and the player is called PVPlayer. [51]

[60] Wideband Code Division Multiple Access
[61] Radio Link Control

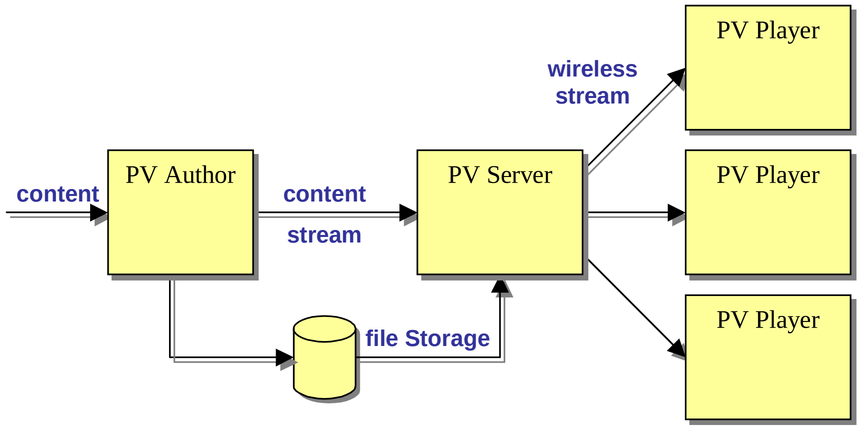


Figure 4.8: The PacketVideo streaming platform

The player makes a request for a compressed video file that resides on the streaming server. Upon client request, the server directs a video file to the client by sending the file using socket-communication. Before sending the video file into the network, the file is segmented, and the segments are typically encapsulated with special headers appropriate for streaming traffic. There are many options for delivering the video from the server to the application. [51]

The file can be sent over UDP at a constant rate equal to the drain rate at the receiver. As soon as the client receives compressed video from the network, it decompresses it and plays it back. [51]

But the client-application could also have a few seconds playout delay in order to eliminate network-induced jitter. The client accomplishes this by placing compressed media that it receives from the network into a client buffer as shown in figure 4.9. Once the client has got a few seconds of buffer it begins to drain it. For this the fill rate $x(t)$ is equal to the drain rate d, except when there is a packet loss, in which case $x(t)$ is momentarily less than d. [51]

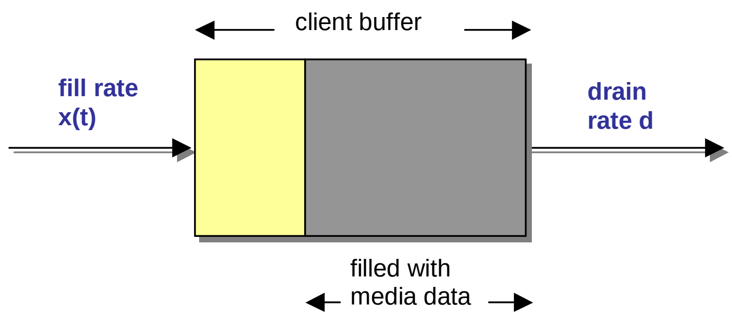


Figure 4.9: Client buffering [51]

A third option is to send the media over TCP. The server pushes the media file into the TCP socket as fast as it can. The client reads from the TCP socket as fast as it can and places the compressed media into the buffer, just as in the second case. The difference is that TCP retransmits lost packet and the fill rate *x(t)* fluctuates with time due to TCP congestion control and window flow control. Therefore, for this case, the behavior of *x(t)* will very much depend on the size of the client buffer. If the buffer is large enough to hold the entire media file, then TCP will make use of all of the instantaneous bandwidth available so that *x(t)* can become much larger than *d*. If, on the other hand, the client buffer is small, then *x(t)* will fluctuate around the drain rate *d*. Risk of client starvation is much larger in the latter case. [51]

A fourth possibility is to make use of the scalability option in the MPEG-4 standard and deliver UDP packets with variable data rate. This could be done in the streaming video platform from PacketVideo that was used in the mVideo prototype. The component called PVAuthor can encode both for variable and constant rate. A component in the server called FrameTrack makes variable bitstreams possible through dynamic rate control that adjusts the frame rate in the presence of channel fluctuations and bit errors. This is done to optimize the user experience. [33]

The PVPlayer monitors important networks parameters and conveys these to PVServer with periodic RTP/RTCP reports. PVServer uses algorithms, which increase or decrease the frame rate in response. The player supports playing back dynamically scaled MP4 source. [33]

Intervals for bandwidth, frame rates etc. are specified in the PVAuthor when encoding the content. It is possible to encode for many variable bandwidth scenarios. By reducing rebuffering and packet losses, FrameTrack provides, according to PacketVideo, the best- possible video streaming experience under constraints of available dynamic user bandwidth. Figure 4.10 shows how the PacketVideo platform uses dynamic rate control to deal with variable bandwidth. [33]

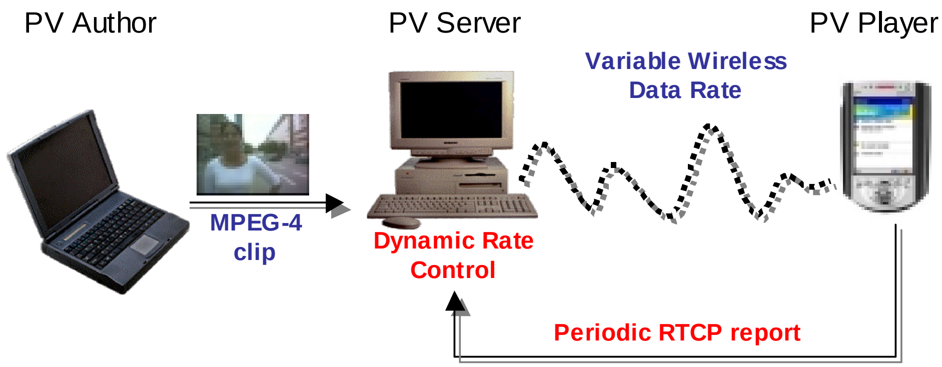


Figure 4.10: The PacketVideo platform can handle variable bandwidth scenarios by using dynamic rate control [33]

This example with the PacketVideo platform ends this section. In the next section we try to summarize this first part of chapter 4 by looking at some advantages and disadvantages with streaming.

4.1.7 Pros and cons with streaming

We start with looking at three benefits with streaming:

- One of the most obvious benefit is the shorter downloading times. With streaming, you do not have to wait until the entire file downloads but can start to watch after just a couple of seconds.
- Since users do not get a copy of the file, publishers and content providers can protect copyright.
- The use of UDP packets and retransmission mechanisms in link layer makes the network throughput more efficient. Retransmission of lost TCP packets over wireless networks is not suitable for streaming.

Now we go on with looking at some of the drawbacks:

- Error-free delivery cannot be guaranteed because slow Internet connections or Internet traffic can cause breaks in listening.
- Since users do not get a copy of the file, the file can not be locally stored for off-line viewing.
- The slower the Internet connection; the poorer the quality. [25]

4.2 Image coding and data compression

Here starts the second part of chapter 4. This part deals with image coding and data compression, which is vital for streaming.

4.2.1 Information theoretic concepts

Let us return to the Shannon-model introduced in the beginning of this chapter. A simplified version is shown in figure 4.11.

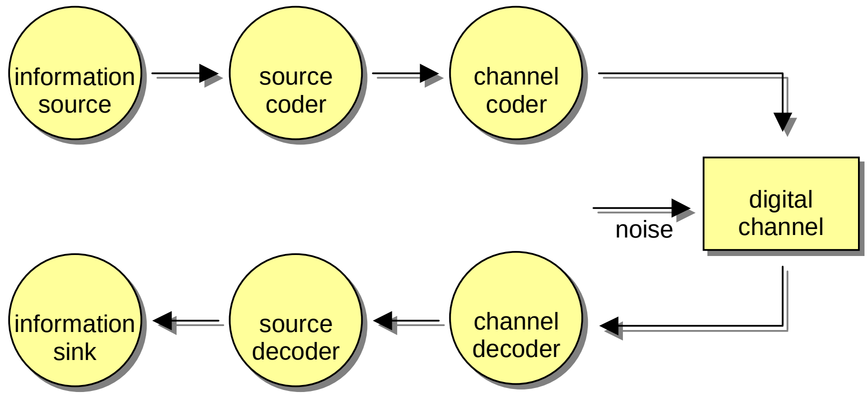


Figure 4.11: Communication system model [50]

The concepts presented by Shannon constitute the base of modern Information Theory. The information source may represent an analog signal, such as a video signal or discrete symbols, e.g. generated by a keyboard. The source coder transforms the source message into strings of binary data (compression). The channel coder represents the binary data in a form suitable for transmission over the channel, taking noise and other disturbances into account (by the use of FEC and interleaving). The channel decoder and source decoder act so as to generate an acceptable estimate of the source message.

There may be several reasons for coding. One could be to yield a compact representation of the source symbols (source coding). Another reason could be to use codes tolerant towards errors in those cases where the source symbols are to be transmitted over a noisy channel (channel coding). Whatever reason one may have to do the coding, it is usually required that the code is uniquely decodable. By this is meant that the original source symbols must be possible to extract from the codewords. [50]

Compression of multimedia content is necessary for transmission. There are many compression systems but they all have that in common that they require two algorithms: one for compressing the data at the source, and another for decompressing it at the destination. These algorithms are referred to as the encoding and decoding algorithms, respectively. [52]

These algorithms have certain asymmetries. Usually, a multimedia document will only be encoded once but will be decoded thousands of times. This means that the encoding algorithm is allowed to be slow and require expensive hardware whilst the decoding algorithm must be fast and cheap. This is not the case, though, for all applications. For real-time multimedia, slow encoding is unacceptable. [52]

A second asymmetry is that the encoding/decoding processes do not need to be invertible. It is usually acceptable to have the video signal after encoding and decoding to be slightly different than the original. When the decoded output is not exactly equal to the original input, the system is said to be lossy otherwise it is called lossless. [52]

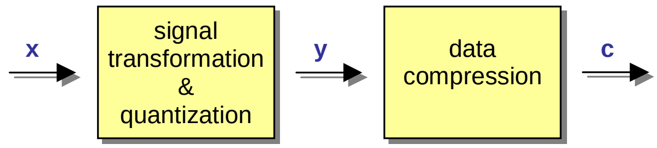


Figure 4.12: A typical image coding process [50]

A typical image coding process is shown in figure 4.12. The input *x* represents the image signal (e.g. a video signal). This signal is generally a time- and amplitude-continuous signal. Through signal transformation (sampling) and quantization an output *y* is generated. The data compression step generates the information bits suitable for transmission or storage. [50]

An example of a type of compression scheme is source encoding. Source encoding takes advantage of properties of data to produce more compression. The compression is usually lossy. The compression scheme can for example take advantage of patterns in the signal and describe frequently occurring events efficiently. It can also introduce acceptable deviations and for example remove information that the humans cannot perceive. One kind of lossless source coding technique is entropy coding. Entropy coding just manipulates bitstreams with regards to what the bits mean. [52]

4.2.2 Transform coding

The discovery of the Fast Fourier Transform (FFT) algorithm in 1965 opened up several new fields where computation of spectra could be applied. For images in particular it was now possible to apply a Fourier transform to the whole image, do modifications when it comes to frequencies and then, by inverse transform, observe the effect on the image. In 1968 a technique for data compression of images based on the Fourier transform was found. It turned out that most of the frequency components, particularly the high-frequency ones, had very low amplitudes. Forcing these to zero did not significantly alter the image after inverse transformation. They showed that compression ratios of more than 10:1 could be achieved this way. [50]

Figure 4.13 shows the principle of transform encoding in one dimension. The spatial domain corresponds to the image, where the a-axis is the pixel values and the x-axis is the position along a row or column.

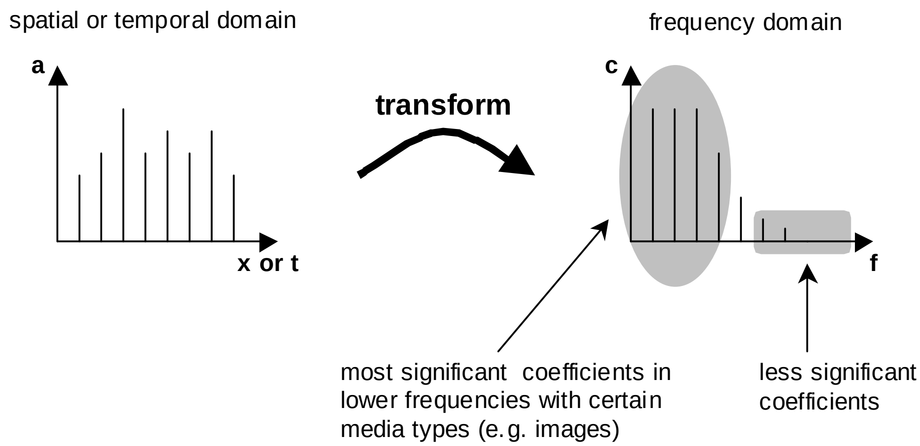


Figure 4.13: Principle of transform encoding

A lot of research has been done to find good transforms functions and the most popular transform today is the discrete cosine transform (DCT). The DCT transforms a signal or image from the spatial domain to the frequency domain. With an input image, A, the coefficients for the output "image," B, are:

$$B(k_1,k_2) = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} 4 \cdot A(i,j) \cdot \cos\left[\frac{\pi \cdot k_1}{2 \cdot N_1} \cdot (2 \cdot i + 1)\right] \cdot \cos\left[\frac{\pi \cdot k_2}{2 \cdot N_2} \cdot (2 \cdot j + 1)\right]$$

The input image is N2 pixels wide by N1 pixels high. A(i,j) is the intensity of the pixel in row i and column j; B(k1,k2) is the DCT coefficient in row k1 and column k2 of the DCT matrix. The DCT input is an 8 by 8 array of integers. This array contains each pixel's gray scale level; 8 bit pixels have levels from 0 to 255. The output array of DCT coefficients contains integers; these can range from –1024 to 1023. For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT. The lower right values represent higher frequencies, and are often small, small enough to be neglected with little visible distortion.

Described in the preceding paragraph is also the use of block transform coding. There are advantages in dividing up the image in smaller subimages (blocks) since it is easier to implement methods for transform coding (the storage requirements become smaller and the computations become simpler). This also opened up the possibility to treat the blocks differently thus adapting to the local statistics of the image. The principle of block-wise transform coding is shown in figure 4.14. [50]

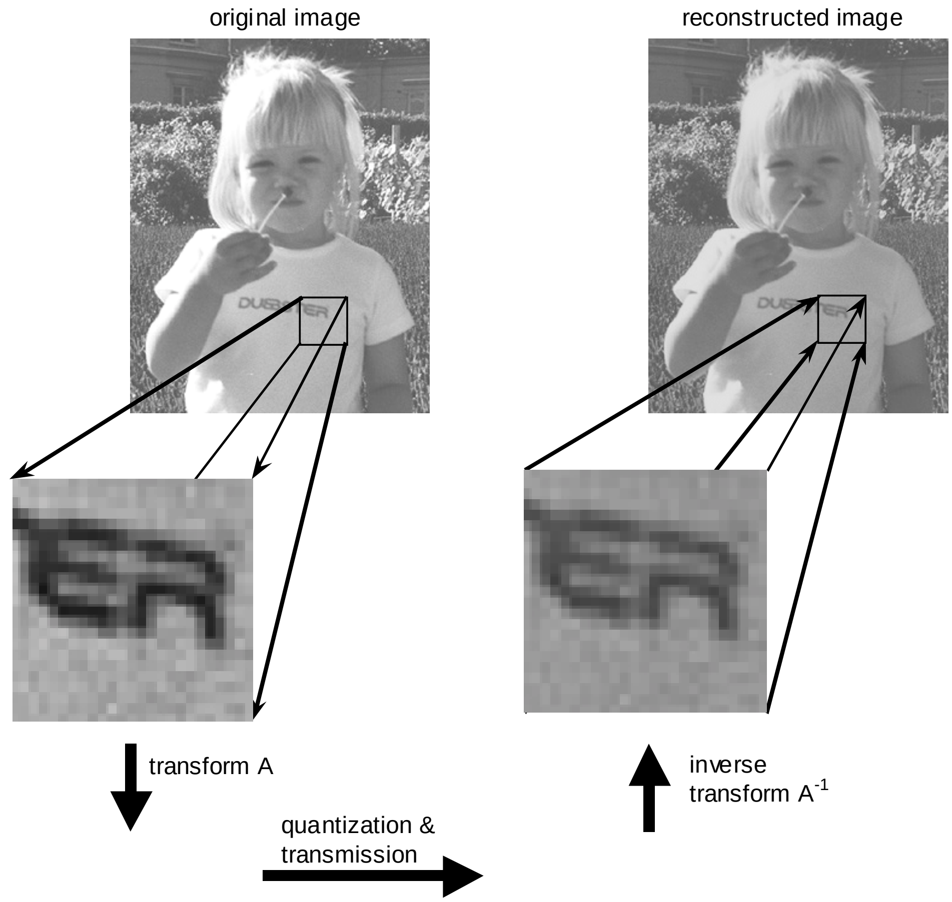


Figure 4.14: Principle of block-wise transform coding

4.2.3 Motion estimation and compensation

Successive video frames may contain the same objects (still or moving). Motion estimation examines the movement of objects in an image sequence to try to obtain vectors representing the estimated motion.

Motion compensation uses the knowledge of object motion to achieve data compression. Motion estimation and compensation have become powerful techniques to eliminate the temporal redundancy due to high correlation between consecutive frames.

In real video scenes, motion can be a complex combination of translation and rotation. Such motion is difficult to estimate and may require large amounts of processing. However, translational motion is easily estimated and has been used successfully for motion compensated coding.

Most of the motion estimation algorithms make the following assumptions:

- Objects move in translation in a plane that is parallel to the camera plane, i.e. the effects of camera zoom, and object rotations are not considered.
- Illumination is spatially and temporarily uniform.
- One object blocking another and uncovered background are neglected.

One technique of motion estimation is to use a block-matching algorithm (BMA). BMAs estimate motion on the basis of rectangular blocks and produce one motion vector for each block.

In a typical BMA, each frame is divided into blocks, each of which consists of luminance[62] and chrominance[63] blocks. Usually, for coding efficiency, motion estimation is performed only on the luminance block. Each luminance block in the present frame is matched against candidate blocks in a search area on the reference frame. These candidate blocks are just the displaced versions of original block. The best candidate block is found and its motion vector is recorded.

Motion estimation is used to predict a block of value in the next picture using a block in current picture. The location difference between these blocks is called Motion Vector, and the difference between these two blocks is called prediction error. This information helps to avoid redefining blocks that do not change but only move from frame to frame. Consequently the motion vector and the resulting error can be transmitted instead of the original luminance block; thus inter-frame redundancy is removed and data compression is achieved.

When a decoder obtains this information, it can use this information and the current picture to reconstruct the next picture. This process is usually called motion compensation. In general, the motion compensation is the inverse process of motion estimation. Figure 4.15 illustrates a process of block-matching algorithm.

[62] Luminance is the relative brightness that controls the red, green and blue proportions in a picture.
[63] Chrominance is the color information - red, green and blue - that is added to the luminance signal to create a color picture.

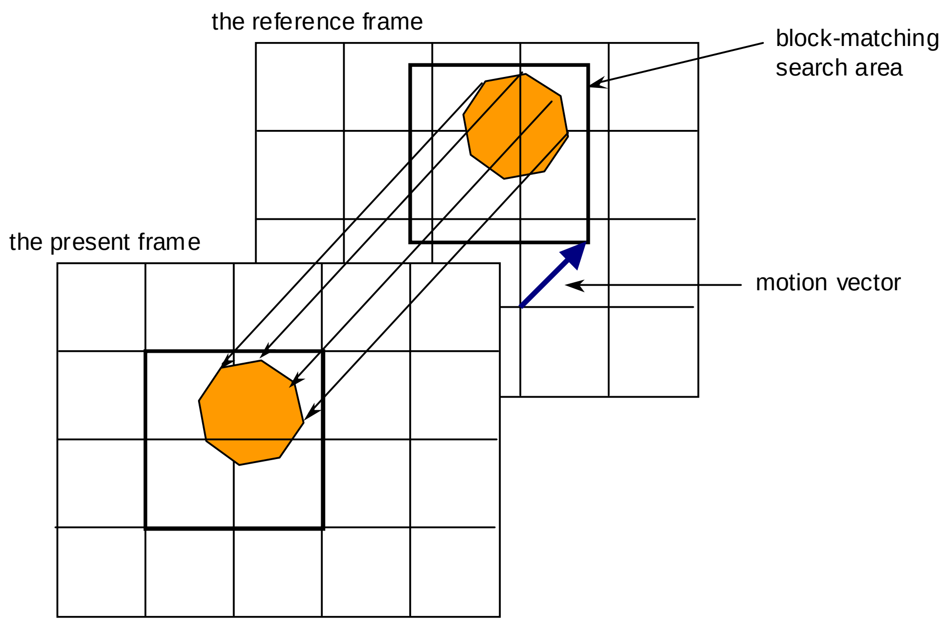


Figure 4.15: Process of block-matching algorithm

Transform encoding, motion estimation and motion compensation are tools for image coding and data compression. [11] In the next section we describe compression schemes that use these tools.

4.2.4 Introduction to MPEG

MPEG (Moving Picture Experts Group) is the ISO committee, which is responsible for defining the various MPEG video specifications. MPEG-1, originally defined in 1992 was aimed at full screen video stored on CD-ROM. It has since been incorporated into Video CD. MPEG-2 came later and was intended for digital television applications and also used for DVD-Video. MPEG-3 was intended for HDTV but this was later incorporated into MPEG-2. [10]

MPEG-4 follows the successful MPEG-1 and MPEG-2 but blends the audio and video compression technologies of its predecessors with those of the web and 3D graphics. This means extending the object-based interactivity to audio and video, adding efficiency and real-time behavior to scene description, and allowing combinations of natural and synthetic content. The challenge is to avoid the duplication of contents for different platforms (TV, wireless, Internet). [46]

Sequences of MPEG video comprise three different types of frames. These are:

- **I-Frames** (Intracoded): Encoded only to compress a single frame without reference to any other frame in the sequence.

- **P- frames** (Predictive): Encoded as difference from last I- or P-Frame and predicting value for each new pixel.
- **B-Frames** (Bidirectional): Encoded as differences with the last or next I- or P-Frame to predict the previous or next I- or P-Frame.

Figure 4.16 shows an example of an I-, P-, and B-Frame sequence in MPEG-2 (MPEG-1 use two B-Frames between P-Frames).[10]

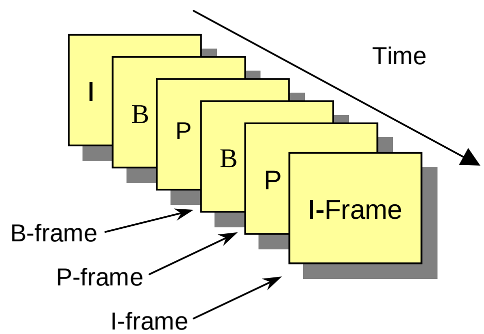


Figure 4.16: Frame types

I-Frames use DCT to compress a single frame without reference to any other frame in the sequence.

P-Frames are coded as difference from the last I- or P-Frame. By dividing the picture in macroblocks (16x16 pixels) and searching for changes this is achieved. P-Frames make use of motion prediction and DCT encoding. As a result P-Frames will give a compression ratio better than I-Frames but depending on the amount of motion present.

B-Frames are coded as differences from last or next I- or P-Frame and use both motion prediction and DCT encoding. B-Frames use prediction as for P-Frames but for each block either the previous or the next I- or P-Frame. This gives improved compression compared with P-Frames, because it is possible to choose for every macroblock whether the previous or next frame is taken for comparison. [10]

The I-Frames need to be periodically distributed. If all frames depend on their predecessors any end-receiver that misses a frame (e.g. due to packet loss) could never decode any subsequent frames. [52]

4.2.5 MPEG-4

The MPEG standard used in the mVideo prototype is MPEG-4. The MPEG-4 standard provides core technologies for efficient storage, transmission and manipulation of video data in a wide range of storage and transmission media. [12]

In addition to the Internet, the MPEG-4 standard is also designed for low bit-rate communications devices, which are sometimes wireless. But whether wired or not, devices can have differing access speeds depending on the type of connection and traffic. In response, MPEG-4 supports scalable content; it allows content to be encoded once and automatically played out at different rates with acceptable quality for the communication environment at hand. [24] The MPEG-4 visual standard has been explicitly optimized for three bitrate ranges: below 64 kbit/sec, 64-384 kbit/sec, and 384-4000 kbit/sec [12].

The MPEG-4 standard itself only defines a bitstream syntax and the decoder semantics. In essence, the standard tells an implementer how to construct a bitstream so that any compliant decoder can decode it. The decoder implementation need to correctly interpret the meaning of bits and render the associated image in order to be a "compliant" decoder. [34]

The implementer can add value in the specific properties of the bitstream decoder and post-processing, which are outside the scope of the standard. An encoder implementation can be differentiated from other implementations through the algorithms used for motion estimation, rate control, and pre-processing, to name a few. [34]

Structure and syntax. In MPEG-4 the audio and video components are known as objects. These can exist independently, or multiple ones can be grouped together to form higher-level audiovisual bonds. The strength of this so-called object-oriented approach is that the audio and video can be easily manipulated. [24]

A video object may consist of one or more layers to support scalable coding. The scalable syntax allows the reconstruction of video in a layered fashion starting from a standalone base layer, and adding a number of enhancement layers. This allows applications to generate a single MPEG-4 video bitstream for a variety of bandwidth and/or computational complexity requirements[64]. [12]

Visual objects in a scene are described mathematically and given a position in a two- or three-dimensional space. Similarly, audio objects are placed in a sound space. When placed in 3-D space, the video or audio object need only be defined once; the viewer can change his viewing point, and the calculations to update the screen and sound are done locally, at the user's terminal. This is a critical feature if the response is to be fast and the available bit-rate is limited, or when no return channel is available, as in broadcast situations. [24]

An MPEG-4 visual scene may consist of one or more video objects. Each video object is characterized by temporal and spatial information in the form

[64] A special case where a high degree of scalability is needed, is when static image data is mapped onto two or three dimensional objects. To address this functionality, MPEG-4 provides a special mode for encoding static textures using a wavelet transform.

of shape, motion, and texture. An MPEG-4 visual bitstream provides a hierarchical description of a visual scene as shown in figure 4.17. [12] The four different levels are:

- **Visual Object Sequence (VS):** The complete scene that may contain any 2-D or 3-D natural or synthetic objects and their enhancement layers. [12]
- **Video Object (VO):** A video object corresponds to a particular object in the scene. In the simplest case this can be a rectangular frame, or it can be an arbitrarily shaped object corresponding to an object or background of the scene. [12]

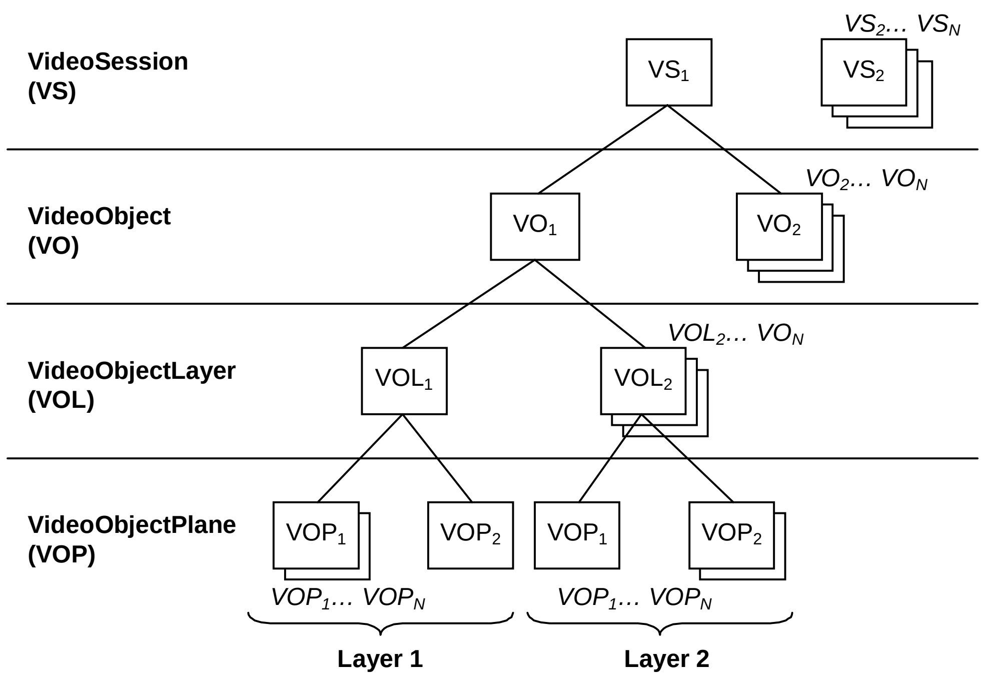


Figure 4.17: MPEG-4 Video syntax structure [12]

- **Video Object Layer (VOL):** Each video object (VO) can be encoded in scalable (multi-layer) or non-scalable form (single layer) represented by the video object layer (VOL). The VOL provides support for scalable coding. A video object can be encoded using spatial or temporal scalability, going from coarse to fine resolution. [12]
- **Video Object Plane (VOP):** A VOP is a time sample of a video object (VO). VOPs can be encoded independently of each other, or dependent on each other by using motion compensation. A conventional video frame can be represented by a VOP with rectangular shape. There are

three modes of coding VOPs; these are I-, P-, and B-VOPs (see figure 4.18). The three modes correspond exactly to I-, P-, and B-Frames described in previous section, i.e. I-VOPs are coded without any information from other VOPs. P- and B-VOPs are predicted based on I- or other P-VOPs. [12]

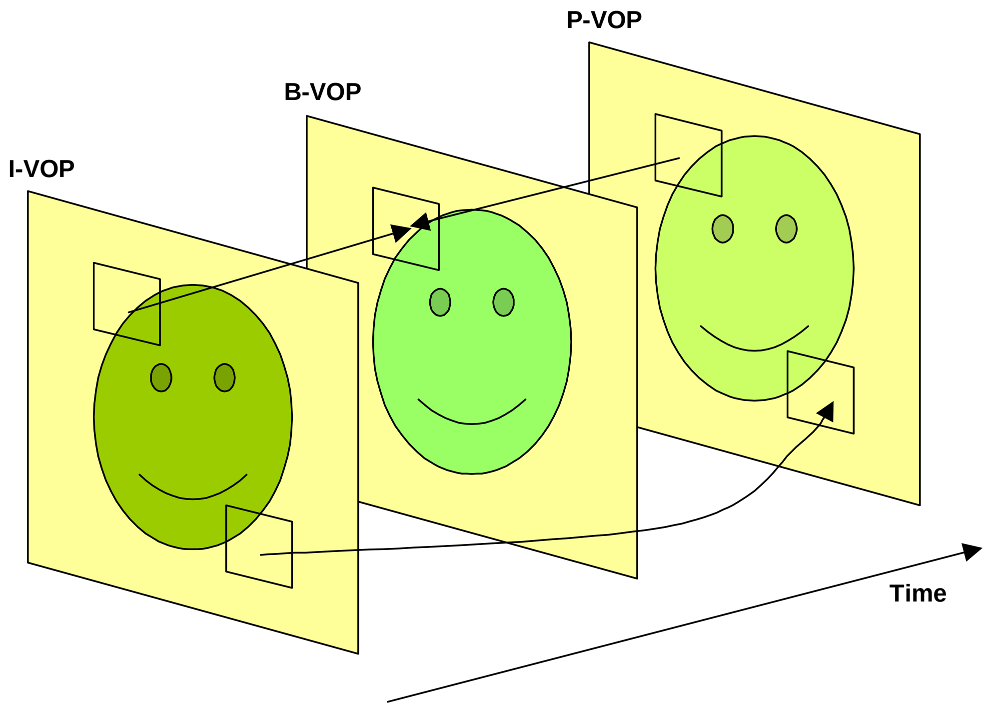


Figure 4.18: The three modes of VOP coding. [12]

A video object plane can be used in several different ways. In the most common way the VOP contains the encoded video data of a time sample of a video object. In that case it contains motion parameters, shape information and texture data. These are encoded using macroblocks. It can also be used to code a sprite. A sprite is a video object that is usually larger than the displayed video, and is persistent over time. There are ways to slightly modify a sprite, by changing its brightness, or by warping it to take into account spatial deformation. It is used to represent large, more or less static areas, such as backgrounds. Sprites are encoded using macroblocks. [12]

Figure 4.19 illustrates the general block diagram of MPEG-4 encoding and decoding based on the notion of video objects. Each video object is coded separately.

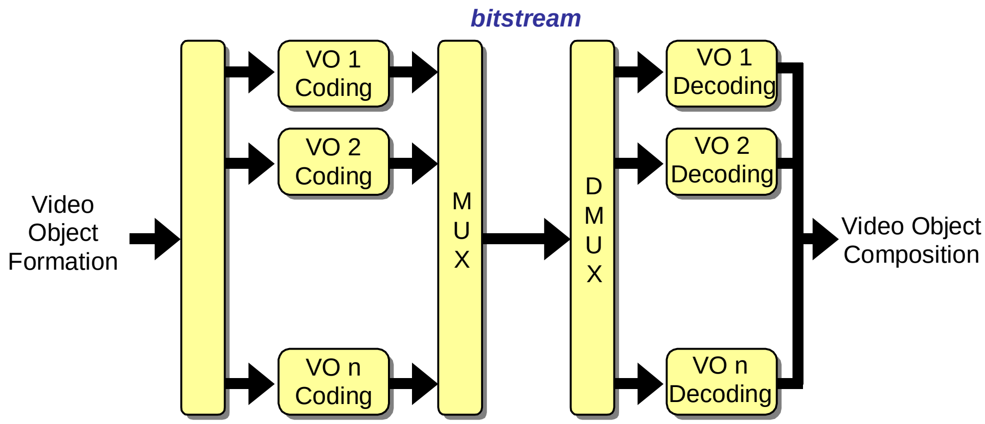


Figure 4.19: General block diagram of MPEG-4 video [12]

A visual scene may consist of one or more video objects. MPEG-4's language for describing and dynamically changing the scene is named the Binary Format for Scenes (BIFS). BIFS commands are available not only to add objects to or delete them from the scene, but also to change visual or acoustic properties of an object. MPEG-4 uses BIFS for real-time streaming, that is, a scene does not need to be downloaded in full before it can be played, but can be built up on the fly. [24]

No specific transport mechanism is defined in MPEG-4. Existing transport formats and their multiplex formats suffice, including for example ATM[65], and RTP on the Internet. But to remedy matters, a small tool in MPEG-4, FlexMux, was designed to act as an intermediate step to any suitable form of transport. FlexMux describes how elementary audio and video streams may be grouped and identified. The TransMux layer is an abstraction outside the scope of MPEG-4 Systems that describes the transport mechanism used to deliver MPEG-4 data over a channel. Examples of the relation between FlexMux and TransMux methods are shown in figure 4.20. In the figure, the FlexMux groups data into Adaptation Layer (AL) Protocol Data Units (PDU) before transmitting with the TransMux. [34]

[65] Asynchronous Transfer Mode, a network technology that supports realtime voice and video as well as data.

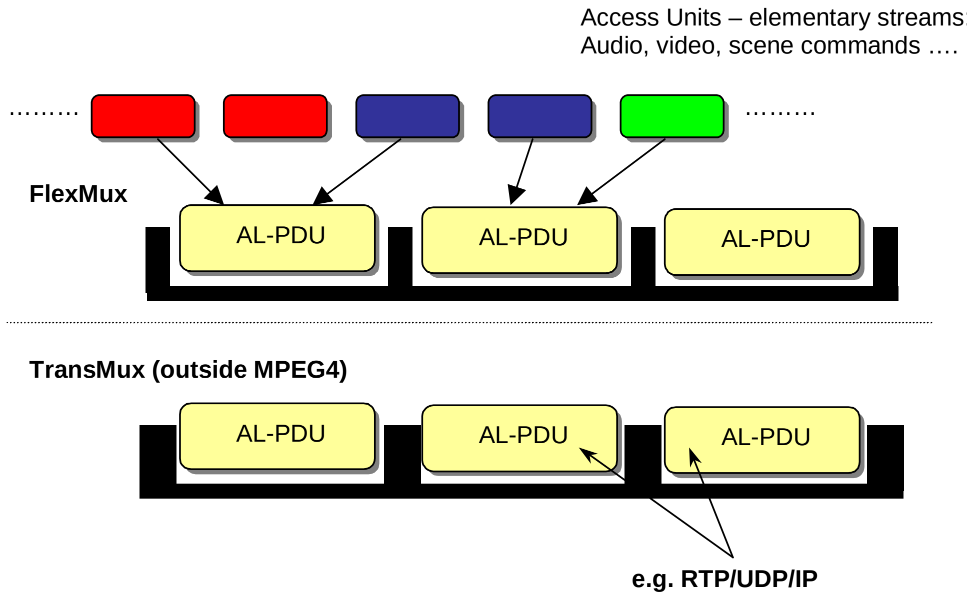


Figure 4.20: MPEG-4 FlexMux defines how to group and identify streams, TransMux describe the transport mechanism to use. [34]

The play-out of the multiple MPEG-4 objects is coordinated at a layer devoted solely to synchronization. Here, elementary streams are split into packets, and timing information is added to the payload of these packets. These packets are then ready to be passed on to the transport layer. [12]

Timing information for the decoder consists of the speed of the encoder clock and the time stamps of the incoming streams, which are relative to that clock. Two kinds of time stamps exist: one says when a piece of information must be decoded; the other says when the information must be ready for presentation. The distinction between the types of stamp is important. In many video compression schemes, some frames are calculated as an interpolation between previous and following frames. Thus, before such a frame can be decoded and presented, the one after it must be decoded (and held in a buffer). For predictable decoder behavior, a buffer model in the standard augments the timing specification. [12]

MPEG-4 Visual Profiles. The MPEG-4 standard addresses both compression of natural and synthetic video scenes. The standard can be separated into profiles and levels, where the predominant visual profiles for natural video content are:

- The Simple Visual Profile, which provides efficient, error resilient coding of rectangular video objects, especially applicable to wireless video communication and transmission. [34]
- The Scalable Visual Profile, which adds support for coding of temporal and spatial scalable objects to the Simple Visual Profile. This is

especially appropriate for delivering video over multiple types of communication channels that may have variable throughput, such as wireless data channels. [34]

- The Core Visual Profile, which adds support for coding of arbitrary-shaped and temporarily scalable objects to the Simple Visual Profile. The addition of arbitrary-shaped objects makes the Core Visual Profile much more complex to implement than the previous profiles. [34]
- The Main Visual Profile, which adds support for coding of interlaced, semi-transparent, and sprite objects to the Core Visual Profile. This profile can be used for broadband entertainment quality video. [34]
- The N-Bit Visual Profile, which adds support for coding video objects having pixel-depths ranging from 4 to 12 bits to the Core Visual Profile. This profile is useful for MPEG-4 applications tied to cameras with extended pixel depths, such as infrared cameras. [34]

The PacketVideo platform used in the mVideo prototype supports the Simple Visual Profile and the Scalable Visual Profile. These are suitable to use for wireless applications because of computational resources and data rates associated with wireless handsets today. [34] Next we take a look at the mechanisms within these two profiles that provides efficient, error resilient coding, and temporal and spatial scalable objects.

Error resilience. The error resilience functionality is important for universal access through error-prone environments, such as mobile communications. MPEG-4 provides several mechanisms to allow error resilience with different degrees of robustness and complexities. These mechanisms are offered by tools providing means for resynchronization, error detection, data recovery and error concealment. There are four error resilience tools in MPEG-4 visual: resynchronization, data partitioning, header extension code, and reversible variable length codes. [12]

- **Resynchronization:** This is the most frequent way to bring error resilience to a bitstream. It consists of inserting unique markers in the bitstream so that in the case of an error, the decoder can skip the remaining bits until the next marker and restart decoding from that point on. MPEG-4 allows for insertion of resynchronization markers after an approximately constant number of coded bits (video packets). [12]
- **Data partitioning:** This method separates the bits for coding of motion information and those for the texture information. In the event of an error, more efficient error concealment may be applied when for instance the error occurs on the texture bits only, by making use of the decoded motion information. [12]
- **Header extension code:** These binary codes allow an optional inclusion of redundant header information, vital for correct decoding of video. This way, the chances of corruption of header information and complete skipping of large portions of bitstream will be reduced. [12]

- **Reversible Variable Length Codes (RVLC):** These RVLCs allow to further reducing the influence of error occurrence on the decoded data. RVLCs are codewords that can be decoded in forward as well as backward manners. In the event of an error and skipping of the bitstream until the next resynchronization marker, it is possible to still decode portions of the corrupted bitstream in the reverse order to limit the influence of the error. Figure 4.21 shows data recovery with the video packet using possible RLVC. [12]

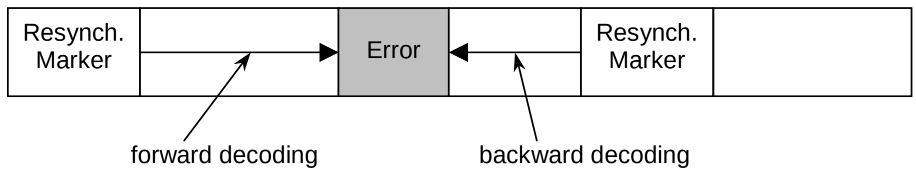


Figure 4.21: Data recovery [12]

Scalability. Spatial scalability and temporal scalability are both implemented using multiple VOLs. Consider the case of two VOLs: the base-layer and the enhancement-layer. For spatial scalability, the enhancement-layer improves the spatial resolution of a VOP provided by the base-layer. Similarly, in the case of temporal scalability, the enhancement-layer may be decoded if the desired frame rate is higher than that offered by the base-layer. Thus, temporal scalability improves the smoothness of motion in the sequence.

MPEG-4 uses a generalized scalability framework to enable spatial and temporal scalabilities. This framework allows the inclusion of separate modules, as necessary, to enable the various scalabilities. As shown in figure 4.22, a scalability preprocessor is used to implement the desired scalability. It operates on VOPs.

For example, in the case of spatial scalability, the preprocessor down-samples the input VOPs to produce the base-layer VOPs that are processed by the VOP encoder. The midprocessor takes the reconstructed base-layer VOPs and up-samples them. The difference between the original VOP and the output of the midprocessor forms the input to the encoder for the enhancement layer.

To implement temporal scalability, the preprocessor separates out the frames into two streams. One stream forms the input for the base-layer encoder, while the other is processed by the enhancement-layer encoder. The midprocessor is bypassed in this case. [12]

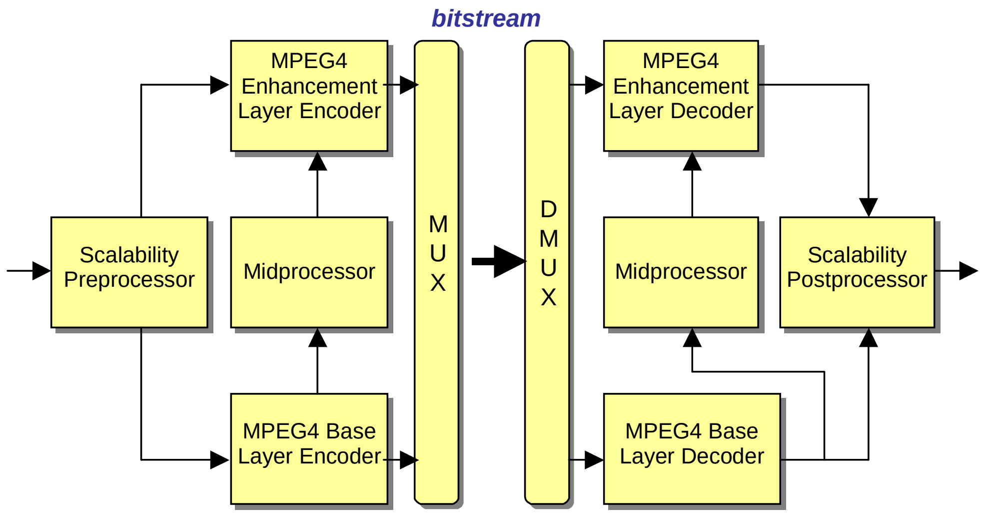


Figure 4.22 Block diagram of MPEG-4 generalized scalability framework. The specific algorithms implemented in the preprocessor, midprocessor and postprocessor depend upon the type of scalability being enabled. [12]

If a VOP in the enhancement-layer is temporarily coincident with an I-VOP in the base-layer, it could be treated as a P-VOP. VOPs in the enhancement-layer that are coincident with P-VOPs in the base-layer could be coded as B-VOPs. Since the base-layer serves as the reference for the enhancement layer, VOPs in the base-layer must be encoded before their corresponding VOPs in the enhancement layer. Figure 4.23 illustrates an example of how the enhancement layer can be decoded from the base layer using spatial scalability. [12]

In temporal scalability, the frame rate of the visual data is enhanced. Enhancement-layers carry information to be visualized between the frames of the base-layer. The enhancement layer may improve the resolution of only a portion of the base-layer, or the enhancement-layer may improve the resolution of the entire base-layer. Figure 4.24 shows base and enhancement layer behavior for temporal scalability the former case. [12]

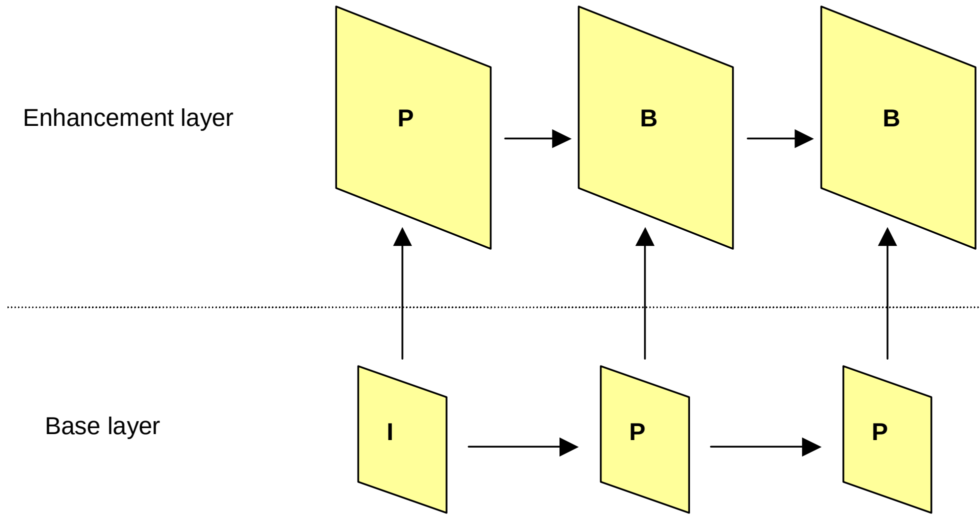


Figure 4.23: Illustration of base and enhancement layer behavior in the case of spatial scalability [12]

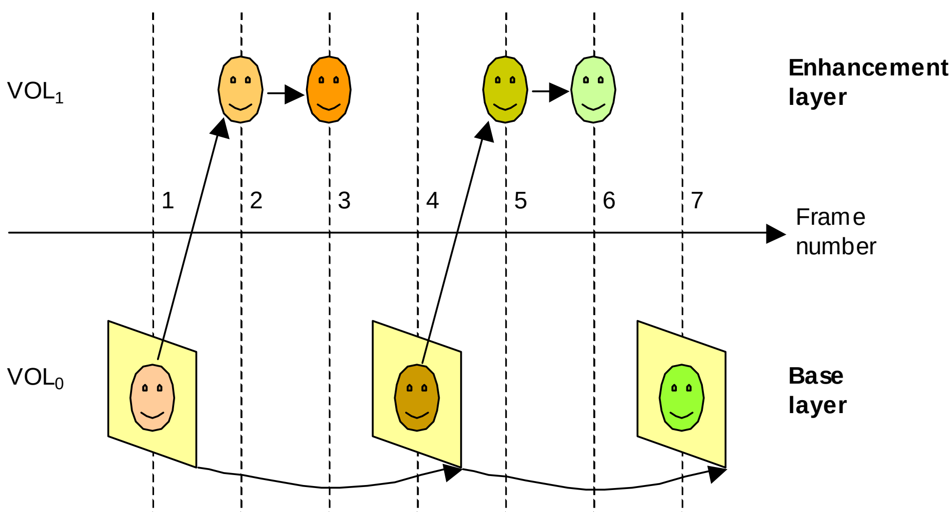


Figure 4.24: Base and enhancement layer behavior for temporal scalability, when improving only a portion of the base-layer [12]

Rate control. An important conformance point is the maximum bit-rate, or the maximum size of the decoder bit stream buffer. Therefore, while encoding a video scene, rate control and buffer regulation algorithms are important building blocks in the implementation of the MPEG-4 video standard. In a variable bit-rate (VBR) environment, the rate control scheme attempts to achieve optimum quality for the decoded scene given a target bit-rate. [12]

In constant bit-rate (CBR) applications, the rate controller has to meet the constraints of fixed latency and buffer size. To meet these requirements, the rate control algorithm controls the quantization parameters. The MPEG-4 video standard describes a possible implementation of a rate control algorithm. The algorithm uses a Scalable Rate Control scheme (SRC) that can satisfy both VBR and CBR requirements. The SRC is based on the assumption that rate-distortion function can be modeled by the following equation:

$$R = \frac{X_1 S}{Q} + \frac{X_2 S}{Q^2}$$

R = Rate
X_1 and X_1 = modeling parameters
Q = Quantization parameters
S = Measure of activity in the frame

This scheme achieves frame rate control for both CBR and VBR cases. A rate control algorithm for multiple video objects is derived from this algorithm, by using a bit allocation table based on Human Visual Sensitivity (HVS) of color tolerance. This results in a bit allocation for each object. Next, the numbers of coded bits per block are estimated based on block variance classification, and a bits estimation model. This results in a reference quantization parameter. The object is encoded using this reference parameter. While encoding the object, small adjustments to the reference parameter can be made depending on the possible deviation of the predicted bits from the actual bits. If the number of actual bits produced is much higher than the allocated bit budget, a frame skip parameter is computed that allows skipping several instances of the video object to reduce the current buffer level. [12]

4.2.6 PacketVideo MPEG-4 visual algorithms

To end this second part of chapter 4 we take a look at what tools from the MPEG-4 standard that is used in the PacketVideo (PV) platform. The PV platform supports the *Simple Visual Profile* and the *Scalable Visual Profile* in the MPEG-4 standard, which means supporting rate control, error resilience and pre/post-processing.

Rate Control. The PVPlatform supports bit-streams that are both variable rate and constant rate. This is achieved by using multiple layer bit-streams of temporal scalability. PacketVideo uses this technique to enable the delivery of video over highly variable rate wireless channels. This methodology, known as FrameTrack, utilizes an estimate of the channel bandwidth computed from the RTCP reports transmitted between a server and client. Given an estimate of the instantaneous channel bandwidth the server can deliver a combination of base layer and enhancement layer frames to achieve the desired channel throughput. The decoder can then decode any frames received and continue decoding without experiencing buffer underflow problems. [34]

Error resilience. The PVPlatform supports the use of resynchronization markers to deal with the containment and concealment of errors. The platform also makes use of data partitioning, which implies that the motion information is separated from the texture information. When errors are detected in the texture component, only the bits associated with the motion vectors are used to create the decoded image. When errors are detected in the motion data, PV algorithms tries to recover lost motion vector information. This recovered information is then used for concealment. This “motion-compensated concealment” approach results in decoded video that hides serious artifacts from the viewer. This technology is known as SignalTrack in the PV platform. [34]

Processing. Post-processing is critical to the subjective quality of decoded video, especially at lower bit-rates. Two of the principal artifacts observed in digitally compressed video are blocking artifacts and ringing artifacts. These artifacts are a result of coarse quantization of the DCT coefficients associated with macroblocks in the picture. The removal of high frequency data results in ringing at the edges of moving objects and visible distortions along regular block intervals. In order to combat these problems in a low complexity manner, PV algorithms attack the filtering problem adaptively. By choosing which pixels to filter, and which to skip, the post-processed scene can be much more effectively rendered, without substantial extra computational resource consumption. One requirement of the decoder is that it must be able to post-process and render each frame of video before it is time to process the next frame, leading to a real-time constraint on these algorithms. [34]

Figure 4.25 ends this chapter; this figure shows general encoding and decoding diagrams for the PV platform. The figure summarizes where in the encoding-decoding process the different tools and techniques within the MPEG-4 standard is used. It is worth noting the similarity to the communication system model given in the beginning of this chapter.

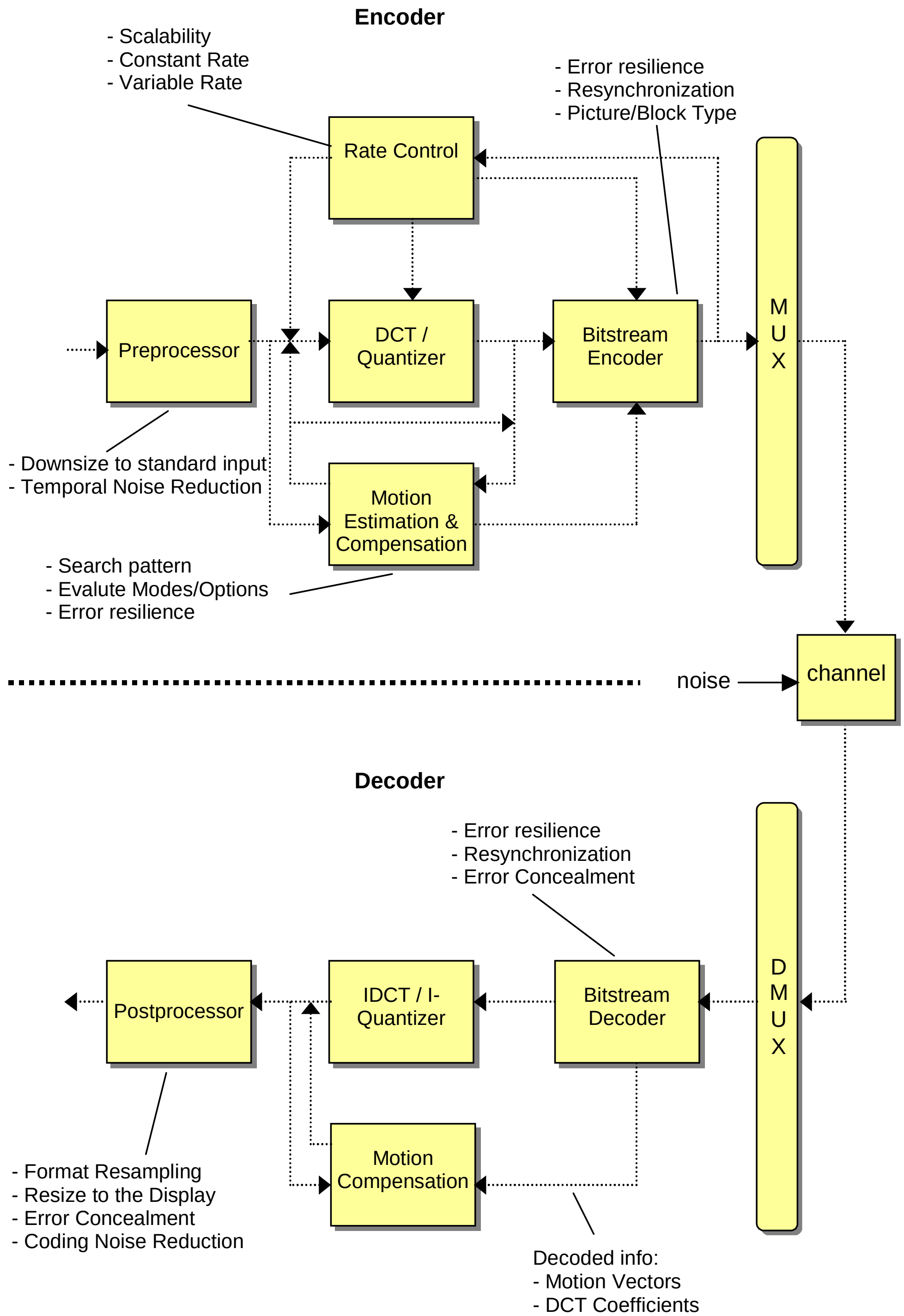


Figure 4.25: PacketVideo encoder/decoder diagram [34]

4.3 Conclusions

Chapter 4 deals with the third issue for the thesis – to present problems and technical solutions that are associated with video streaming.

It proved that parameters that affected the video quality could be divided into two categories that could be connected to the communication systems model by Claude Shannon. The first category derives from networks and the second from image coding and data compression.

One of the corner stones when communicating over the Internet is the TCP/IP reference model. The TCP/IP model is a “best-effort” service. That means that it does not make any guarantees of delays to applications. This could be a problem when it comes to multimedia applications including, for example, streaming. There are three major problems for multimedia applications when sending data over the Internet. These are packet loss, end-to-end delay and delay jitter. Packet loss results in some packets never reaching its destination. This loss of information could be helped with retransmission mechanisms, forward error correction or interleaving. The end-to-end delay is the accumulation of transmission processing and queuing delays in routers, propagation delays, and end-system processing delays along a path from source to destination. A delay may make packets arrive in wrong order. This is called delay jitter. Delay jitter is not acceptable for a multimedia application, but adding sequence numbers to the packets could help this problem.

In order to better accommodate multimedia traffic new protocols that better handles multimedia are added. These protocols are Real Time Streaming Protocol (RTSP), Real Time Transport Protocol (RTP) and Real Time Control Protocol (RTCP). These protocols provide, amongst others, functions for monitoring packet losses and re-ordering packets according to sequence numbers. They also implement a sort of “remote control” and add functions such as pause/resume, fast forward and other features often wanted when using multimedia.

A last thing when considering the channel coding part of Shannons systems model is the technique used for streaming. There are several techniques to choose from. One is to send a file over UDP at a constant rate equal to the drain rate at the receiver. A second variant is to make use of a buffer and let the client have a few seconds of playout delay. Another way of doing it could be to use TCP and a buffer (in order to avoid packet loss). Finally, there is a possibility to use the scalability option in the MPEG-4 standard and deliver UDP packets with variable data rate. The choice of technique has a large impact on the behavior of streamed material, e.g. a buffer much decreases the risk of client starvation.

The MPEG-4 standard mentioned above, is an image coding and data compression scheme that is developed to amongst others suit streaming media over low bandwidth networks. The use of a good compression scheme is vital since the channel often is limited and the performance is dependent on the amount of data that has to be transferred.

In MPEG-4 the audio and video components are known as objects and one video object may consist of one or more layers to support scalable coding. The MPEG-4-standard can be separated into five profiles. The two profiles supported by PacketVideo are the Simple Visual Profile and the Scalable Visual Profile. The scalability in itself can be done in two different ways - spatial scalability and temporal scalability. Both of these techniques are implemented using multiple layers. PacketVideo uses temporal scalability.

Another important feature in MPEG-4 is the rate control. With the rate control, bitstreams with both variable rate and constant rate can be used. PacketVideo uses this technique to enable the delivery of video over wireless channels with highly variable bandwidth.

5 PVPlatform evaluation

This chapter deals with the final issue for the thesis – to study and evaluate the chosen platform through different channels. This has been achieved by testing the platform in an emulation[66] environment called NIST Net Network Emulator. This emulator made it possible to emulate different kind of network situations. The tests presented in this chapter are not made in order to evaluate the mVideo Messaging prototype in any way. These tests are strictly concerning the behavior of the streaming video platform.

Reading chapter 4 before reading this chapter is recommended since it otherwise can be hard to understand the performed tests and the discussions about the observations. The following issues and network situations have been studied.

- **Encoding settings** (section 5.2). There are many opportunities when encoding the video signal. In this test we evaluate how different encoding-settings affect the performance of the PacketVideo platform.
- **Influence of network bandwidth** (section 5.3). How does the platform handle networks with different bandwidth and how is the buffering time affected by the bandwidth?
- **Effect of the rate control** (section 5.4). How effective is the rate control with temporal scalability? We compare scalable mode and non-scalable mode in different situations.
- **Effect of packet drops** (section 5.5). How does the platform handle disturbance in the form of packet drops?
- **Effect of delay jitter** (section 5.6). How does the platform handle disturbances in the form of packet reordering?

5.1 Test environment

The testing has been made using a special evaluation network tool called NIST Net Network Emulator. The NIST Net Network Emulator is a tool for emulating performance dynamics in IP networks. The NIST Net package allows a single Linux PC to be set up as a router between two networks to emulate a wide variety of network conditions.

The tool is designed to allow controlled, reproducible experiments with network performance sensitive/adaptive applications. By operating at the IP level, NIST Net can emulate the critical end-to-end performance

[66] Emulation is testing code inserted into a "live" implementation, allowing the live implementation to emulate (imitate) the performance characteristics of other networks. Simulation is a totally synthetic test environment, without a live component. [30]

characteristics imposed by various wide area network situations or by various underlying subnetwork technologies. [30]

The network environment we used was built up according to figure 5.1.

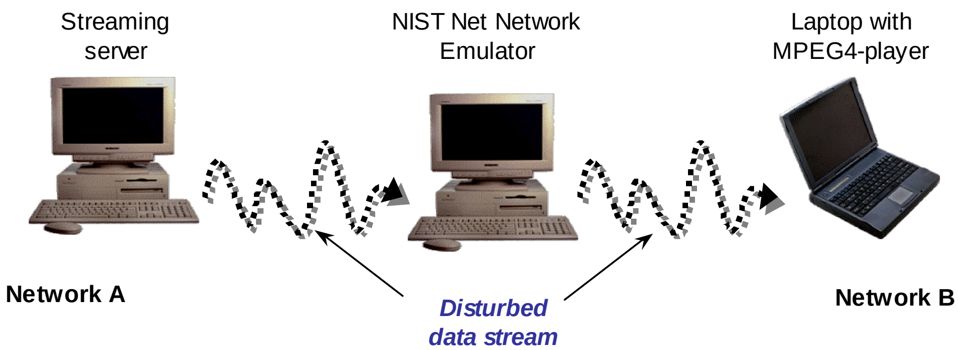


Figure 5.1: The network emulator acts as a router between network A and network B.

As origin video source, a video signal (size 160x120) that was stored in a 91.7 MB avi-file was used. The video sequence is 60 seconds long and has a frame rate of 30 frames per second. The video material does not include audio. This avi-file have been encoded in the PVAuthor to MPEG-4 format and then transferred to PVServer.

All testing have been performed with PacketVideo platform 2.0 that was not commercially released. According to PacketVideo the 2.0 platform is not the latest version available. The test methodology and all results have been performed by the authors only and were not endorsed by PacketVideo. Neither did PacketVideo participate in the testing.

5.2 Test: Encoding settings

In the encoding tool, PVAuthor, there are several types of settings a user can specify:

- **Target bit rate**, the target bit rate is to be set so that it matches the bandwidth of the target network.
- **Scalable or Non-Scalable mode,** in scalable mode the rate control is activated meaning that the server can stream at variable bit rate.
- **Min bit rate,** in scalable mode this specifies the lower limit for the bit rate.
- **Min Frame Rate**, in scalable mode this specifies the lower bounds for frame rate. The min frame rate sets the P-Frame interval for base layer. [47]
- **Max Frame Rate,** in scalable mode this specifies the upper bounds for frame rate. Max frame rate sets the P-base + P-enhanced interval. [47]

- **I-Frame Interval**, defines the frequency that I-Frames will be used in the encoded content.
- **Video Size,** max size is 160x120 pixels

We have looked at how the I-Frame interval and target bit rate affect the picture quality and file size. Regarding the file size, we looked at both the size of the actual file and the amount of data that is streamed through the network.

We have also looked on the difference between encoding in scalable mode and in non-scalable mode. When scalability is enabled, the user takes advantage of a component in the streaming server called FrameTrack. FrameTrack handles variations in the bandwidth within a network. We have also looked on how the bit rate settings influence the file size.

A company that delivers a 3G service would want to have the best effort that can be. Therefore it is necessary to understand the result of the different settings and in what way an administrator can affect the performance.

5.2.1 Purpose

The influence of I-Frame interval is interesting to look at because, if a small I-Frame interval should have a big effect on file size and amount of data through the network, this could imply a very bursty behavior of the data stream. This is also done to see how big difference in size it is between I-Frames and P-Frames.

Next, a file with different bit rate is encoded in scalable mode and then the same file is encoded using non-scalable mode. The file size and network throughput for scalable and non-scalable case is compared. After this, we encode the file with a minimum bit rate that is half of the upper bit rate and then with a minimum bit rate that is very low (6.25 kbps). This is done to see how good the encoder is to adapt the frames in the enhancement layer to the upper bit rate. This is also done to verify our expectations from the theory.

Finally we take a look at how much impact the target bit rate have on the picture quality. From the users point of view it is interesting to see how good quality different network-speeds can offer and therefore screenshots for video sequences encoded for a 14.4 kbps network and 128 kbps network is shown.

5.2.2 Theory and expectations

As described in section 4.2.4, an I-Frame is encoded just like a still picture. Between the I-Frames there could be P- and/or B-Frames, which are encoded as the difference with the last or next I- or P-Frame to predict the previous or next I- or P-Frame. This ought to mean that the I-Frames are larger than the other when it comes to data size. It also means that the larger amount of I-Frames in an encoded video clip, the higher bit rate and therefore higher quality when there is much motion. The expected result is that the I-Frame interval should have a large impact on both the file size and the network throughput.

When considering the difference for scalable and non-scalable encoded files the files ought to have approximately equal file-sizes. In scalable mode some P-Frames are moved to the enhancement layer where they might be encoded with higher quality than in base layer. This could imply a bigger file size, but since the upper bit rate is the same in both cases the total amount of data that should be transferred through the network ought to be the same. And therefore we expect the file sizes to be equal.

5.2.3 Results and discussion

Figure 5.2 shows the file size and amount of data through the network for files encoded with different I-Frame intervals.

The 91.7MB avi-file was encoded in scalable mode with target bandwidth 43 kbps, min bit rate 21.5 kbps, max frame rate 10 fps, min frame rate 1 fps, and the I-Frame intervals varies. The bandwidth of the network was kept at more than 20 Mbps.

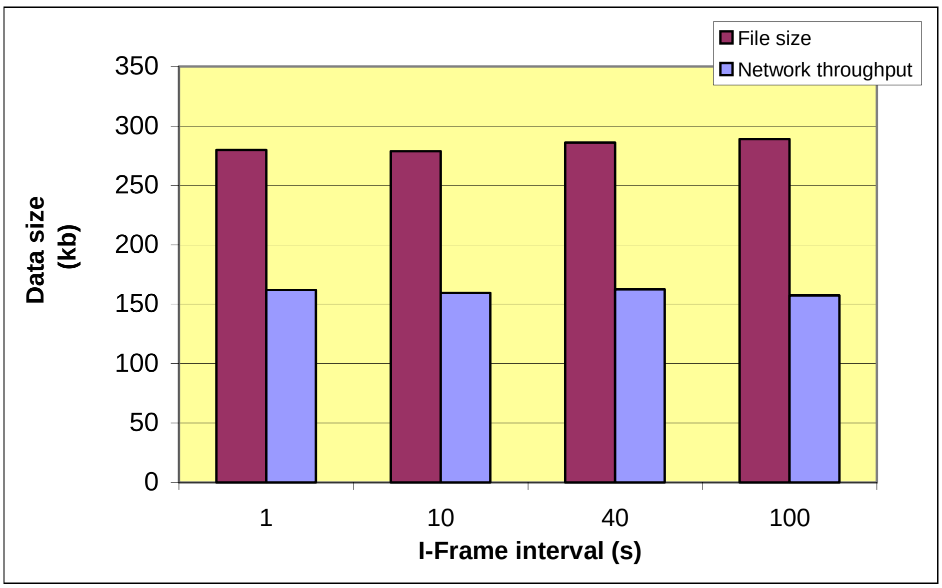


Figure 5.2: Data amount for different I-Frame intervals

As figure 5.2 shows, the I-Frame interval does not seem to have any influence on neither the file size nor the network throughput. In other words, our test does not confirm our expectations from the theory of I-Frames. A surprising observation is the big difference between file size and network throughput.

Through discussions with PacketVideo it has been known that PVPlatform only supports I- and P-Frames[67] [47]. A possible explanation then, could be that P-Frames are just as big as I-Frames. But this is not the fact, the answer lies instead in the fact that PVAuthor tries to maintain a nearly constant bit rate [49]. And when a user tries to put more I-Frames in the bitstream, the encoder will try to maintain the target bit rate by skipping some of the P-Frames and/or reducing the quality of the frames (I- and P-) [49]. In the extreme case, if a user tries to encode excessive amount of I-Frames, all the P-Frames and some I-Frames will be dropped and the amount of data you send through the network will still roughly be the same [49].

Figure 5.3 shows the file size for different bit rates when encoding in scalable and non-scalable mode.

The 91.7MB avi-file with min bit rate 6.25 kbps (for scalable mode), max frame rate 10 fps, min frame rate 1 fps (for scalable mode), and the I-Frame intervals 10 seconds. The target bit rates is varied.

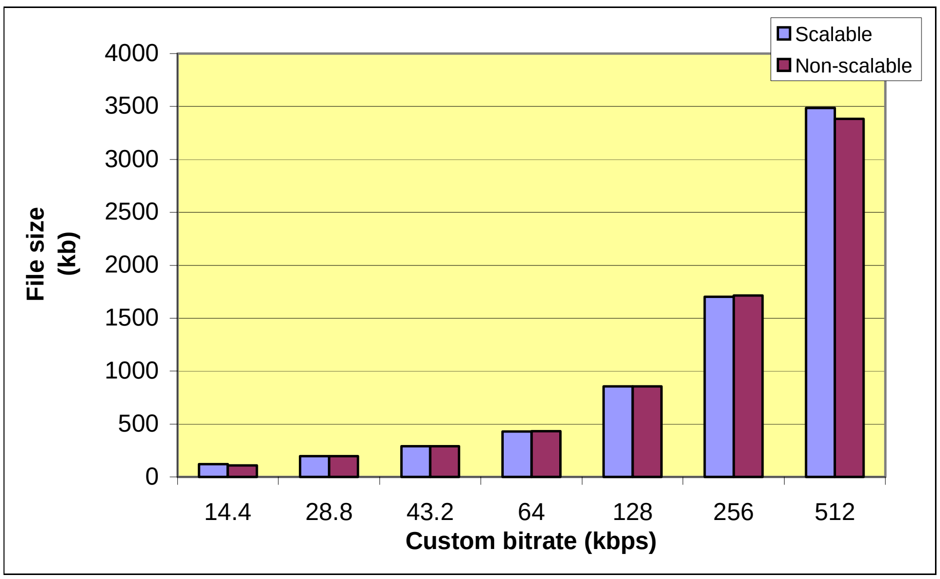


Figure 5.3: File size for scalable and non-scalable encoding

As figure 5.3 shows, there is no observable difference when it comes to the file size. The test confirms the expectations and is an effect of the encoder that tries to maintain a near constant bit rate. When encoding in scalable

[67] According to PacketVideo, B-Frames are not encoded in the base layer because it is not allowed for MPEG-4 Simple Profile compliance. For the enhancement layer, B-Frames are not included for several reasons. One reason is that before MPEG-4 adopted Level 0 of Simple Scalable Profile (a level suitable for wireless streaming), there have been some suggestions about removing B-Frames in order to reduce the complexity. Besides, many of the PacketVideo decoder implementations are on low-power, low-memory embedded systems. [49]

The main disadvantages of B-Frame for streaming is that it slightly increases the end-to-end delay and that the complexity on the client side is slightly higher. However, since the newly adopted Level 0 of SSP in MPEG-4 does support B-Frame, PacketVideo are currently testing and developing our B-Frame solution. [49]

mode the encoder puts some frames in the enhancement layer. These frames could be encoded with higher quality, but the encoder will try to maintain the target bit rate, which means that the throughput and file size will be roughly the same.

Figure 5.4 shows the file network throughput for scalable and non-scalable mode when encoding for different bit rates.

The 91.7MB avi-file was encoded with min bit rate 6.25 kbps (for scalable mode), max frame rate 10 fps, min frame rate 1 fps (for scalable mode), and the I-Frame intervals 10 seconds. The target bit rate is varied. The bandwidth of the network was kept at more than 20 Mbps.

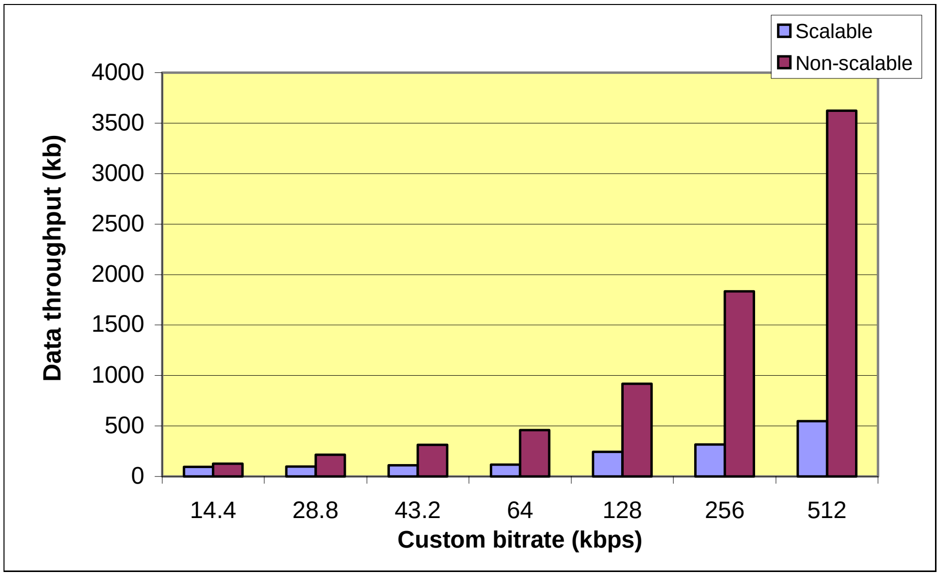


Figure 5.4: Data throughput for scalable and non-scalable model when encoding for different bit rates

The big difference in data throughput is very surprising since the network bandwidth should be big enough not to force the streaming server to lower the amount of data to send for the scalable case.

Through discussions with PacketVideo it has been known that this probably depends on an optimization issue in the streaming server. There are different QoS settings in the streaming server that has to be based on the underlying network. Their explanation is that we do not have the right settings for a high-speed network (20 Mbps) and because of this we see the strange behavior. The conclusion is that these QoS settings are therefore having impact on the observed picture quality.

Figure 5.5 compares files with different range of variable bit rate. One file is encoded with half of the target bit rate as the lower limit and the other is encoded with 6.25 kbps as a constant minimum.

The 91.7MB avi-file was encoded with min bit rate 6.25 kbps and half of the target bit rate for comparison, max frame rate 10 fps, min frame rate 1 fps, and the I-Frame intervals 10 seconds. The target bit rate is varied.

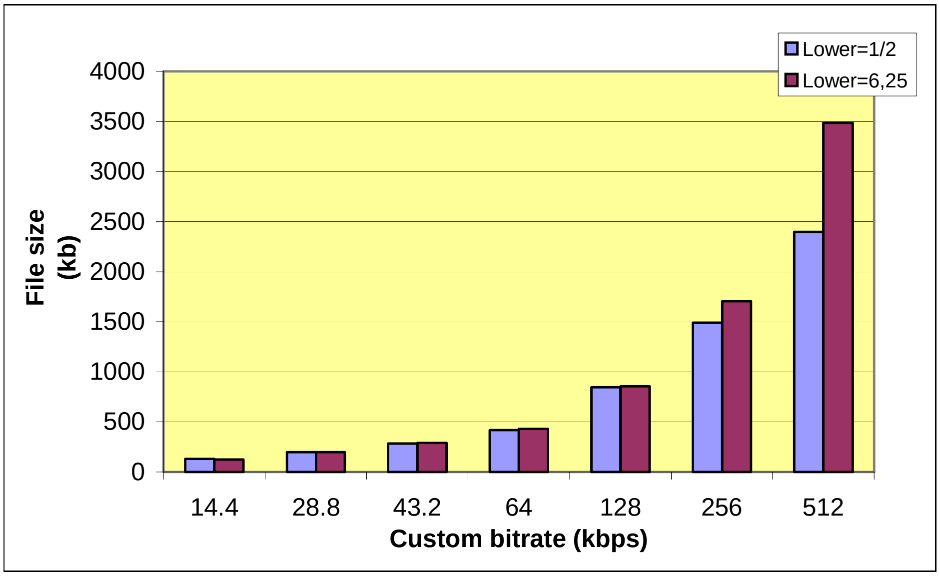


Figure 5.5: File size for different bit rate intervals

At 14.4 kbps the two files do not differ much when it comes to bit rate interval (6.25 vs. 7.2 kpbs as lower limit). Therefore, the files are as large as each other. When increasing the target bandwidth, the difference between the two files increases. The file encoded with 6.25 as lower limit has a rather small, constant amount of base layer frames, while the file encoded with 1/2 of the top value as lower limit increases its amount of base layer frames with the target bandwidth.

The encoded file size is determined by the target bit rate and the total number of frames. Through discussions with PacketVideo it was known that there is some overhead for each video frame stored in the MP4 file. The more frames you have, the higher the overhead will be. Even if two encodes have the same target bit rate, they may not have the same total number of frames. [47]

To explain the observations in figure 5.5 this should mean that the overhead is bigger in the enhancement layer than in the base layer, since there will be more enhancement-layer-frames in the 6.25 case. However, this contradicts with the scalable/non-scalable test above. At that test the size of the files were roughly the same at all bit rates levels. In non-scalable mode the encoder does not produce any enhancement frames, in scalable mode it does.

Also, PacketVideo makes use of some information called hint track, which is used by the server to determine how to put data into packets. The size of the hint track depends on the configuration of the two layers of video, not by the target bit rate. [47] Unfortunately not even this explains the behavior. This to contradicts with the scalable/non-scalable test and therefore we cannot explain the observed result.

Finally, in figure 5.6, screenshots from the player when showing three files encoded for different bit rates are shown.

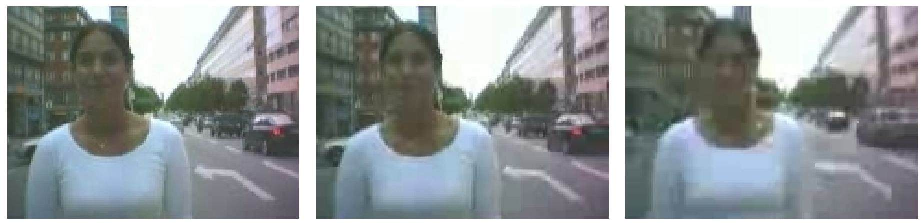


Figure 5.6: Picture qualities for a video sequence encoded for different bit rates

5.3 Test: Influence of network bandwidth

The two variables that have been changed in this experiment are the target bandwidth in PVAuthor and the network bandwidth in the network emulator.

The test has been performed using the avi-file mentioned earlier. This file was encoded at different target bandwidths and streamed through the network emulator at different bandwidth levels. When streaming, the queue length in the network emulator (router) was noted each fifth second. There was no upper limit for the queue length in the router. Noted was also if and when network congestion occurred.

5.3.1 Purpose

This test evaluates the PacketVideo concept when it comes to networks with variable bandwidths. The test tries to answer two questions.

For a given bandwidth-encoded file, how does the channel bandwidth affect network congestion and queue length in the router? That is how good can the streaming server adopt to the current condition in the network with respect to the bandwidth and how long time does it take before the server starts to lower the output data stream. The consequence of a very good adaptability is that one file could be encoded for a large bandwidth spectrum. Which means a large target group with respect to their speed of the connections.

The second purpose of the test is to see how the network bandwidth affects the time to start play video (the buffering time) for PVPlayer?

5.3.2 Theory and expectations

We recall from section 4.1.3, that one of the network problems that could occur was end-to-end delay. In this test the queuing behavior in a router is observed. This information tells how good the server adopts for the current conditions. A small number of packets in the router queue means that the server is sending at a speed that matches the conditions well.

Decreasing the available bandwidth is not equal to an end-to-end delay but from the players point of view the effect could be an insufficient amount of data in the buffer, trigging a network congestion state. Network congestion (or Player rebuffering) occurs when PVPlayer does not have enough "continuous" data in its buffer to render audio + video with respect to real time. From the player’s point of view this means that some packets are delayed since they are not available when needed. From the users point of view this means that the picture is freezing for a while.

When encoding the video, the target bandwidth and the minimum available bandwidth are chosen. These two parameters decide which bandwidth interval the streaming server could vary within. It is hard to predict how big this interval could be, but theoretically the encoder should be able to put a very small part of the frames in the base layer. This should imply a big bandwidth spectrum, on the other hand, it takes some while before the player starts to send good RTCP reports concerning available bandwidth, packet losses etc, back to the streaming server. During this time the server ought to stream at a high level, meaning that the router buffer will be large and the player buffer will be insufficient.

The first thing to happen when starting a streaming session is that the server fills up the player buffer, this takes about seven seconds in the normal case. But the data putted in the buffer is not ought to be adjusted for the current network conditions since it takes a while before a basis for RTCP reports is available. This means that we can expect to see an increasing router buffer queue when the bandwidth spectrum is large. This also means that the buffer time for the player might increase for large spectrum case, since the streamed data is not adopted for current conditions. How long time it takes before the server can adopt will be clear from the tests.

5.3.3 Results and discussion

Figure 5.7 shows the packet-queue in the router. The file is encoded with a target bit rate of 128 kbps and streamed through four different network bandwidths. Figure 5.8 shows the packet-queue in the router when the file is encoded with a target bit rate of 512 kbps.

The 91.7MB avi-file was encoded with min bit rate 6.25 kbps and max bit rate 128 kbps and 512, max frame rate 10 fps, min frame rate 1 fps, and the I-Frame intervals 10 seconds. The network bandwidth is varied.

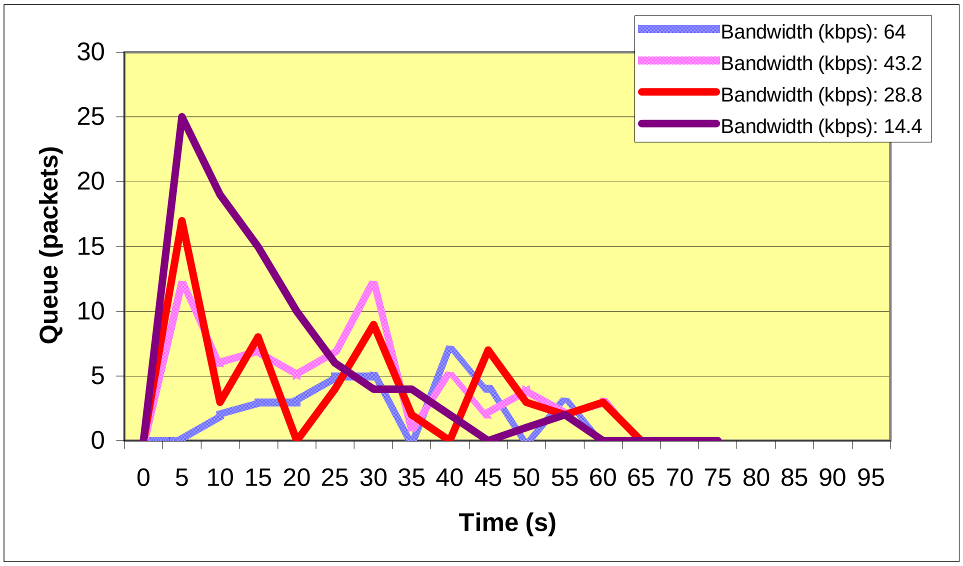


Figure 5.7 Router-queue behaviors at different network bandwidth and encoding with 128 kbps as target bit rate

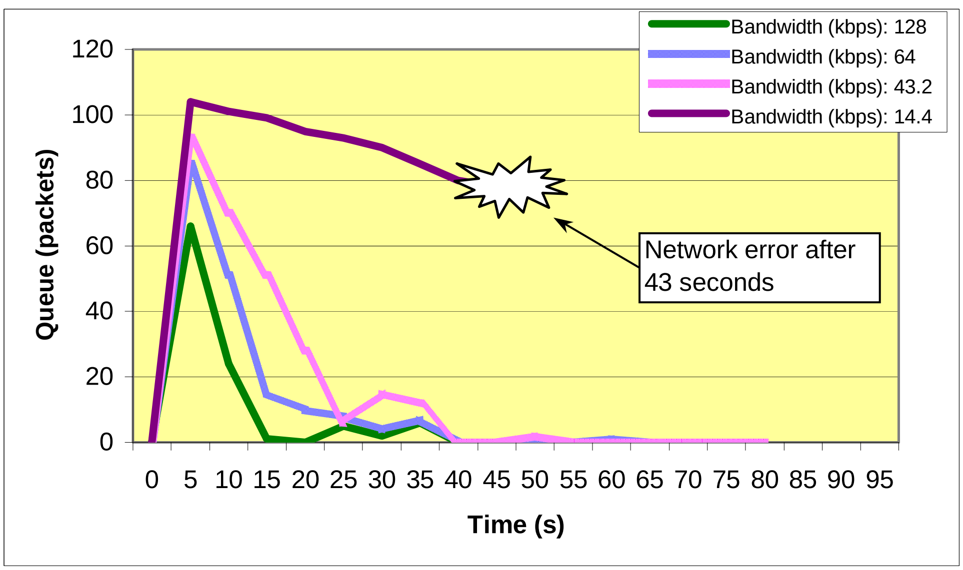


Figure 5.8: Router-queue behaviors at different network bandwidth and encoding with 512 kbps as target bit rate

Figures 5,7 and 5,8 above show on the same behavior, as the bandwidth decreases, more packets are put in to the routers queue just as predicted. A remarkable note of this test is the fact that it was actually possible to stream a file encoded for a 512 kbps target network through a network with a bandwidth of only 43.2 kbps without any network error or congestion. The two figures above also show that it takes about 10 seconds before the server starts to adopt for current network conditions. In the latter case the peak is reached after ten seconds, when there are more than 100 packets in the router queue. In the first case the peak is 25 packets and is also reached after ten seconds. A network error occurs after 48 seconds when streaming the 512 kbps encoded file through a 14.4 kbps network. The error stops the streaming before the intended time.

The conclusion of this test is that it is possible to encode a file for networks with different bandwidths. In the two test presented above, a possibility to encode a file for 512 kbps as maximum target bandwidth, 6.25 kbps as minimum network bandwidth and vary the network bandwidth down to 43.2 kbps without facing network congestion or problems, was shown. Note that no upper limit on the queue length in the router was used.

In the final experiment the buffering times for different channels was investigated. Figure 5.9 shows the buffering times but for a file encoded with a target bit rate of 128 kbps that is streamed through networks with different bandwidths.

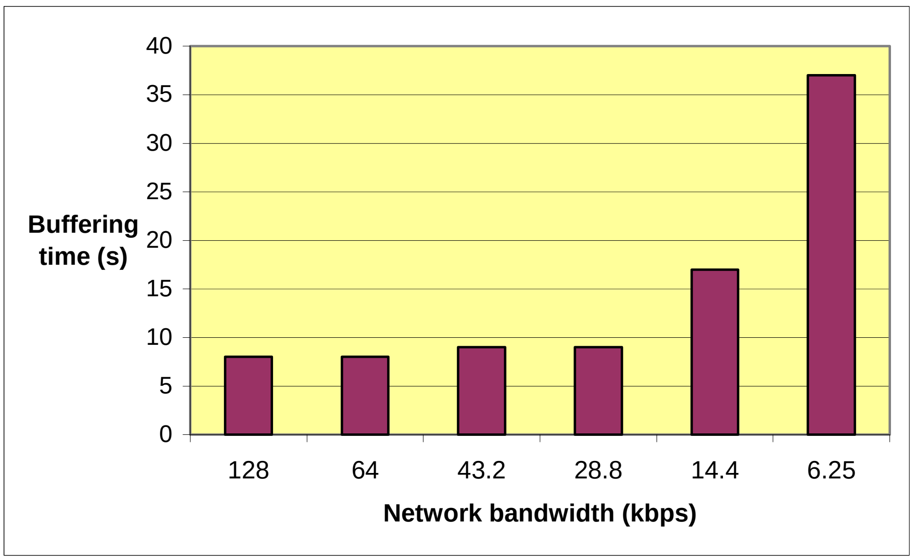


Figure 5.9: Buffering times at 128 kbps target bit rate

Figure 5.9 show that the buffering time does not increase until the bandwidth is lowered under 28.8 kbps. This is surprising since it was expected that a base for the RTCP reports should not be available until after about 20 seconds. The observations could depend on that the player is configured to start play after about seven seconds of buffering, not seven seconds of continuous video. When the available bandwidth was lowered under 28.8 kbps it takes more than seven seconds to fulfill the lower requirements before the player is able to play anything.

5.4 Test: Effect of the rate control

This test is closely linked to the latter test. In this test scalable and non-scalable encoding is compared from the routers point of view. Scalable encoding is much made for networks with varying bandwidths. The bandwidth-test insinuated that the scalable encoding makes it possible to encode for different network bandwidths. But how much does scalability improve compared to a file without scalable possibilities? To answer that is the aim of this test.

The PacketVideo platform comes with a component called FrameTrack. FrameTrack handles variations in the bandwidth within a network. When video material is encoded in PVAuthor, the user can choose to encode in scalability-mode or not. If the scalability is on, FrameTrack is enabled.

The variable that we have been changing is the network bandwidth in the network emulator. We have measured the queue length in router; if/when network congestion occurred and compared the values for a scalable and a non-scalable encoded file. Also the buffering time for the player was noted, that is how long time does it take before the player starts to play the video.

The tests were performed using the same video material as earlier.

5.4.1 Purpose

The purpose of this test is to compare scalable to non-scalable encoding to see what difference it makes regarding congestion time and queue length in router. A big difference implies that the FrameTrack technology is effective and attractive, both for providers and users of the video material. From the providers point of view this means that it is possible to just create one copy of a video clip and still reach users with different connection and still be able to offer as good quality as possible for the current connection. Without the scalability option the provider needs to create one copy for different network speeds, if the streamed quality should be as good as the channel conditions offer, which the user probably wants.

5.4.2 Theory and expectations

The theory and expectations here is much the same as in the previous test. It is expected that the scalable encoded file will better handle low network bandwidths. The risk of having large queues and experiencing network congestion ought to be larger for the non-scalable encoded file.

5.4.3 Results

In figure 5.10 we compare the queue length in the router for two files that were encoded with a target bit rate set to 128 kbps and streamed through a network with a bandwidth of 128 kbps.

In figure 5.11 we compare the queue length in the router for two files that were encoded with a target bit rate set to 128 kbps and streamed through a network with a bandwidth of 64 kbps.

In figure 5.12 we compare the queue length in the router for two files encoded for a 43.2 kbps network, but streamed through a network with a bandwidth of 14.4 kbps.

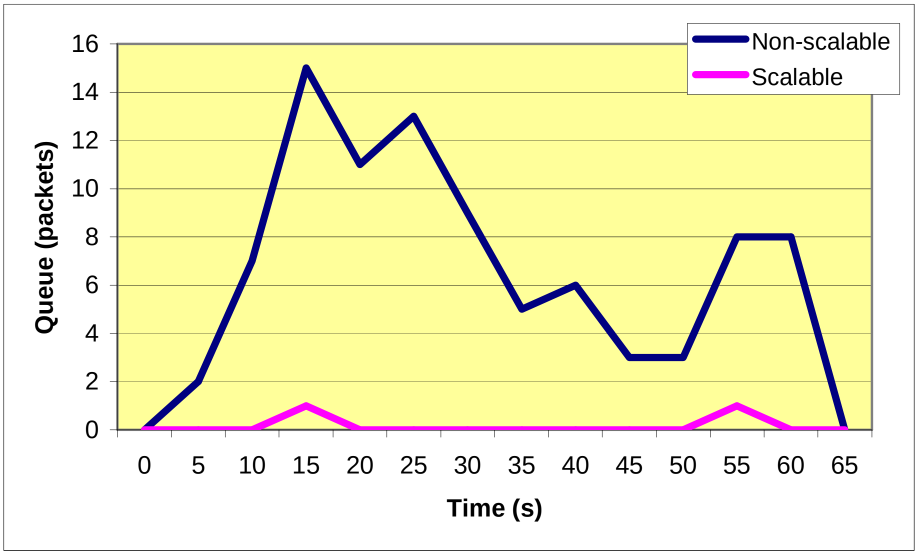


Figure 5.10: Target bit rate of 128 kbps and bandwidth of 128 kbps

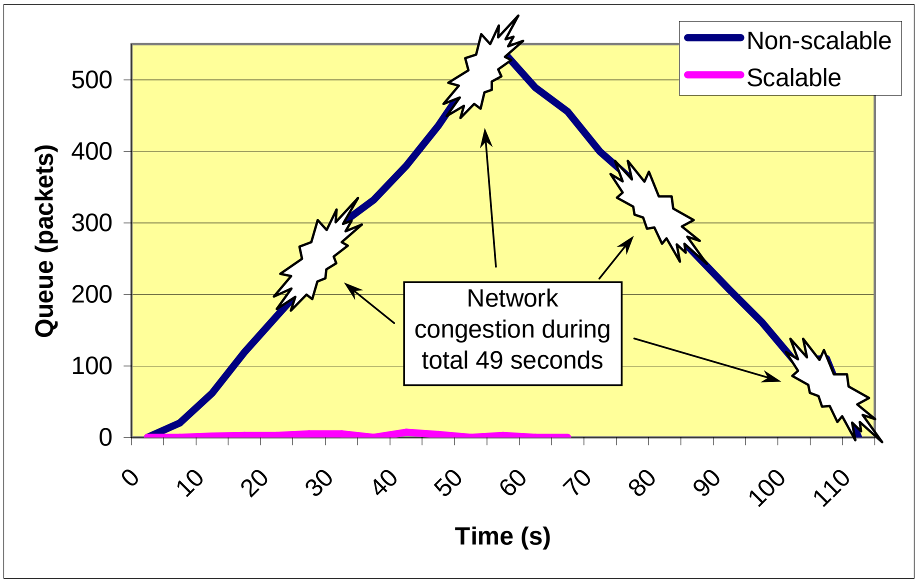


Figure 5.11: Target bit rate of 128 kbps and bandwidth of 64 kbps

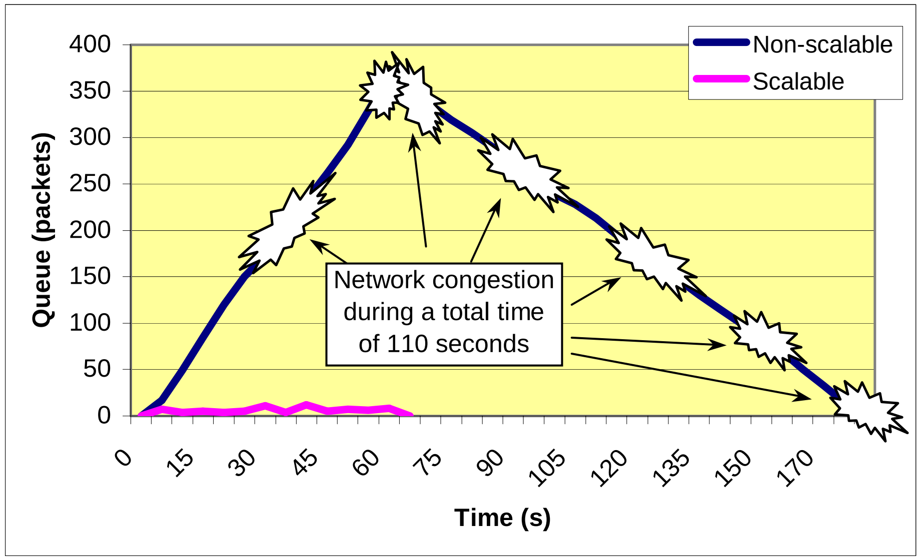


Figure 5.12: Target bit rate of 43.2 kbps and network bandwidth 14.4 kbps

The difference between the scalable and the non-scalable is obvious. Figure 5.10 shows the case when a file encoded with a target bandwidth of 128 kbps was streamed over a channel with 128 kbps as bandwidth; no major difference in the behavior of the packet queue was observed. In figure 5.11 though, where the bandwidth of the channel is only 64 kbps there is a clear difference.

Notable is also that periods of network congestion. 49 seconds and 110 seconds respectively, for the two latter cases. During network congestion the picture freezes.

The conclusion is that the FrameTrack technology, which makes use of scalability, seems to facilitate the streaming.

Finally it was tested how the players buffering time was affected by the scalability option. Figure 5.13 shows the difference in buffering time for scalable and non-scalable files encoded with a target bit rate of 43.2 kbps. The file is streamed through channels with different bandwidths.

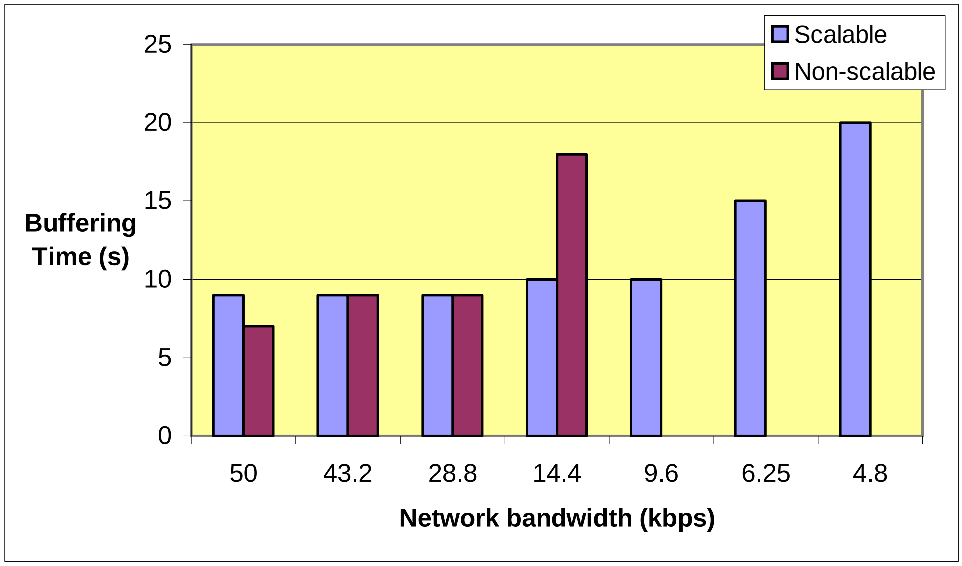


Figure 5.13: Buffering times at 43.2 kbps target bit rate, scalable and non-scalable

A comparison between scalable and non-scalable shows that the buffer time in non-scalable mode increases already at 14.4 kbps, compared to scalable mode where the corresponding value is 6.25 kbps. Notable is also that streaming of the non-scalable file was not possible for network bandwidths of 9.6 kbps and lower.

The buffering time at 14.4 kbps for the scalable file is 10 seconds and in non-scalable mode 18 seconds. The amount of data put in to the player's buffer during the buffering stage, is for the non-scalable file about 260 kb, while the scalable only needs 144 kb to start play.

The conclusion is that the scalability technology decreases the buffering time.

5.5 Test: Packet Drops

This test is performed to measure how the picture quality is affected by packet losses. One way to measure the picture quality would be to achieve some kind of value on the quality (e.g. SNR-value). Since the playing tool PVPlayer does not have a function for saving the streamed material or for picking out one frame out of the stream, another measure has to be chosen. The streamed material has been observed and subjective assessments have be done.

The variables in this test are the network packet loss and network bandwidth that both can be set in the network emulator.

5.5.1 Purpose

This test addresses the problem with packet losses. The test is performed to find out what percentage of the packets that can be lost and how that affects the picture quality. At the edge of where quality gets bad, the packet loss percentage is freeze and the bandwidth is lowered in order to see what happens.

A real situation in which a huge packet loss could be expected, is for example while sitting on a train and the train goes into a tunnel. While the train is in the tunnel the amount of packet loss could be very high. This forces the player to make use of its buffer, which will only last for about six, seven seconds. However, this case is not investigated in this test since that situation would probably force a 100% packet loss per time unit during the tunnel visit. The test performed investigates a 20% packet loss during the whole streaming session.

5.5.2 Theory and expectations

Recall from section 4.1.3 that packet drops occur as IP datagrams are discarded because one or more buffers in the route from sender to receiver are full or when the receiver is not available. This problem is solved in TCP as lost packets are retransmitted. But such retransmission mechanisms are not often used in real-time audio applications because they increase the end-to-end delay. Instead UDP packets are often used and the retransmission is handled at a lower level. The PacketVideo platform relies on the Radio Link Protocol (RLP) with packet retransmission for example, and therefore a 20% packet loss would be considered excessive. [47] In the test environment however this protocol is not available and therefore a 20-percentage packet loss could be forced. This means though, that some of the test values in this test are unrealistic and not representative for a real situation. Still it is interesting to see how the platform handles a high percentage of packet loss.

An interesting issue here is what will happen if some of the packets lost contain I-Frames. If, for example, the first I-Frame is lost, it ought to be hard to see anything until the second comes as the P-Frames are built upon the information in the I-Frames.

5.5.3 Results and discussion

First is investigated how many packets that could be lost before the picture quality became so bad that viewing was no longer possible. This occurred at about a 20 % loss. At 20 % packet loss the bandwidth for the network was decreased. For a 64 kbps network no large deterioration was noticed but by 32 kbps the picture quality became much worse and network congestion was experienced. At 14.4 kbps the quality was very bad and by 9.6 kbps PVPlayer was not able to play the clip.

When streaming, it was noticed that the picture quality became gradually worse for approximately 10 seconds and thereafter markedly better. Figure

5.14 shows estimation of how the picture quality varied with time. The markers on the graph refer to figure 5.15 that are snapshots of the streamed material.

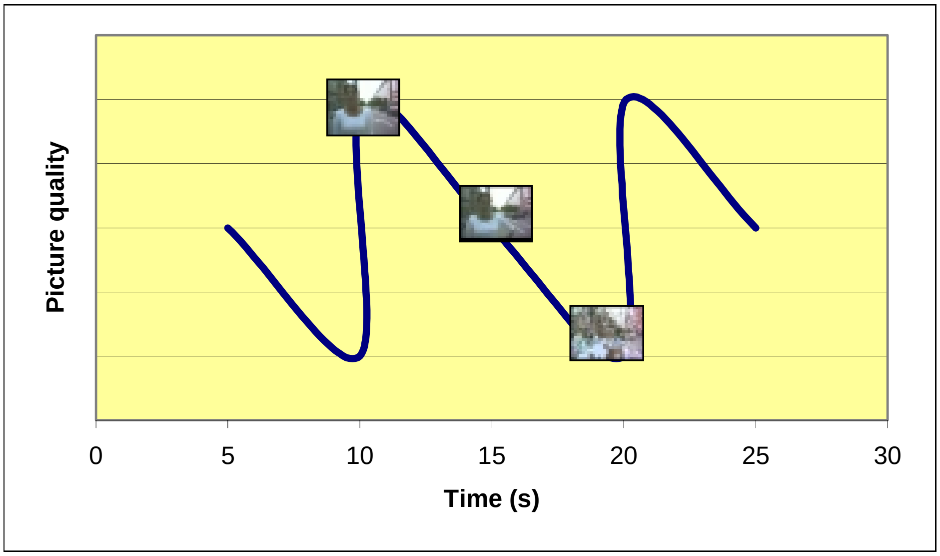


Figure 5.14: Estimation of picture quality, 20% packet loss and I-Frame interval of 10 s.

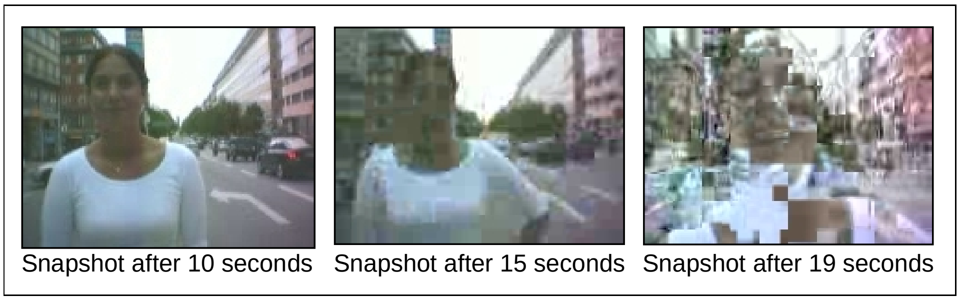


Figure 5.15: Snapshots, 10, 15 respectively 19 seconds after start, 20% packet loss and I-Frame interval of 10 s.

The first file tested was encoded with an I-Frame each 10th second. It was easy to see the effect of the I-Frames; the picture quality grew worse until the appearance of a new I-Frame (approximately each 10th second).

Figure 5.15 first show a snapshot as an I-Frame has arrived. The next picture shows the picture approximately 5 seconds after the I-Frame and the last picture just before the appearance of the next I-Frame.

After the first test another file was encoded with an I-Frame rate of 1 second. This file showed on the same behavior, as in the previous case, but the picture quality never grew so bad as in that case. It was observed that the intervals between picture improvements seemed to be longer than 1 second during this test. This could be due to the loss of I-Frames.

A third test was made with a file encoded with an I-Frame rate of 100 seconds. That means only one occurrence of an I-Frame during the whole session (as the video material is 60 seconds long). Now, the picture quality just grew worse and worse without ever getting better. As discussed in the theory and expectations section earlier, the quality is dependent on whether the I-Frame is lost or not, since the next coming P-Frames uses the I-Frames as reference. In the third case when only one I-Frame could be expected to be available, the lost of this frame ought to have resulted in a total failure. This was never observed, though.

Finally we wanted to see what happened if we froze the packet loss to 20% and then gradually lowered the bandwidth. A file that was scalable encoded with a target bandwidth of 128 kbps was used. At a bandwidth of 64 kbps the result showed no larger deterioration. At 32 kbps the result was noticeable worse, so at 14.4 kbps. At 9.6 kbps PVPlayer was not able to play out the material.

While performing this test it was noted that a high frequency of I-Frames increased the risk of experiencing network congestion. Through discussions with PacketVideo its been known that this phenomena could be explained by the fact that the PVPlatform, depending on the bandwidth and frame rate, is forced to exceed the minimum bit rate [47]. This is due to the fact that each I-Frame will require more bits to encode than a P-Frame, and this constraint may force the targeted minimum kbps to be exceeded [47]. A higher rate of I-Frames will increase the likelihood of player rebuffering (network congestion) [47].

5.6 Test: Effect of delay jitter

With this test we would like to find out how reordering of the packets affects the picture quality. A large normal distribution delay of the packet will reorder the packets and the variable to look at here is therefore the network delay sigma (the variation for the normal distribution) in the network emulator.

Like in the previous test, the picture quality is measured by subjective estimation of the quality.

5.6.1 Purpose

This test addresses the problem with packet delay jitter. We want to find out how large delay that is possible and how such a delay affects the picture quality. At the edge of where quality gets bad, we would like to freeze the delay and decrease the bandwidth in order to see what happens.

5.6.2 Theory and expectations

Packet reordering at the receiver might be the result of the receiver ignoring jitter. Jitter can be removed by using sequence numbers, timestamps or playout delays. Since PacketVideo makes use of the RLP they use sequence numbers in order to avoid jitter. This means that the PacketVideo platform will not try to reorder the incoming packets during this test since the RLP is not available.

The expected effect is that the “old” packets will be discarded, that is if the incoming packet for example has number 3, 6, 4, 5, 8. Packet number four and five will be discarded since packet six already been received. The effect will be much the same as a packet loss.

5.6.3 Results and discussion

First it was studied how much delay that was possible before the picture quality became so bad that viewing was no longer possible. The behavior was similar to the test with packet losses. Sigma equal to one second was ok, and so was two seconds. At three seconds delay congestion was experienced for the first time. By a delay of 15 seconds, the PVPlayer was not able to play the clip.

We stayed at a three seconds delay and decreased the bandwidth. At 64kbps no large deterioration was noticed. By 32 kbps the picture quality was still acceptable but by 14.4 kbps it was very bad and it was hard to see anything.

When increasing the delay, the risk of network congestion increased as well. Network congestion occurs when the PVPlayer does not have enough "continuous" data in its buffer to render audio + video (with respect to real time). Packet loss and delayed I-Frames can potentially force a temporary rebuffering condition when the above condition is triggered.

5.7 Conclusions

Chapter 5 deals with thesis issue four – to study and evaluate the chosen platform through different channels.

In the first test it was investigated how the I-Frame interval and target bit rate affect the picture quality, file size and network throughput. It was observed that the I-Frame interval does not seem to have any influence on neither the picture quality (as long as no packet are lost during the streaming session), file size nor network throughput. Through discussions with PacketVideo it has been known that this depends on the fact that the encoder tries to maintain a nearly constant bit rate. When a user tries to put more I-Frames in the bit stream, the encoder will try to maintain the target bit rate by skipping some of the P-Frames and/or reducing the quality of the frames. In the extreme case, if a user tries to encode excessive amount of I-Frames, all the

P-Frames and some I-Frames will be dropped and the amount of data sent through the network will still roughly be the same.

In another test it was observed that the amount of data sent through a 20 MB network differed very much when using scalability option compared with non-scalable. Through discussions with PacketVideo it has been known that this probably depends on an optimization issue in the streaming server. There are different QoS settings in the streaming server that has to be based on the underlying network. These QoS settings are therefore having an impact on the observed picture quality.

By decreasing the available bandwidth when streaming it was noted that players could get insufficient amount of data in the buffer, trigging a network congestion state. From the users point of view this means that the picture is freezing during the network congestion. But it was possible to stream a file encoded for a 512 kbps target network through a network with a bandwidth of only 43.2 kbps without any network error or congestion. It was also observed that it takes about ten seconds before the server starts to adopt for current network conditions and about seven seconds of buffering time before the player starts to play.

PacketVideo supports scalable technology and the use of this improve the performance compared to not use it. When a 60 seconds video sequence encoded with a target bandwidth of 128 kbps was streamed through a 64 kbps network, network congestion was observed during 49 seconds when the scalability was not used, compared to no congestion when using scalability. Streaming a non-scalable video encoded for a 43 kbps network through a 14.4 kbps network force network congestion of total 110 seconds, compared to no congestion when using scalability option.

It has also been noted that a high frequency of I-Frames increased the risk of experiencing network congestion. Through discussions with PacketVideo it has been known that this phenomena could be explained by the fact that PVPlatform, depending on the bandwidth and frame rate, is forced to exceed the minimum bit rate. This is due to the fact that each I-Frame will require more bits to encode than a P-Frame, and this constraint may force the targeted minimum kbps to be exceeded. A higher rate of I-Frames will increase the likelihood of network congestion.

When dealing with packet loss the effect of I-Frames was observed. When having a constant 20% packet loss during the whole streaming session, the picture quality grew worse until the appearance of a new I-Frame. By setting the I-Frame rate of 1 second it was observed that the picture quality was improved. A large I-Frame interval implied that the picture quality just grew worse and worse during a long time until the next I-Frame was received.

PacketVideo makes use of the RLP to deal with packet loss and delay jitter. RLP was not available during the tests and therefore both packet loss and delay jitter could be investigated. The effect of delay jitter during the tests was very much the same as for packet losses since the PacketVideo platform will not try to reorder the incoming packets.

6 Summary

In this chapter the whole thesis is summarized. The chapter is divided into four parts, the first three deals with the first three issues for the thesis. The last part deals with issue number three and four.

6.1 Present an attractive service

The first issue for the thesis was to present an attractive service for 3G that contains streaming video. The service presented is a Multimedia Messaging Service (MMS). MMS is the ability to send and receive messages that consist of a combination of text, sounds, images and video to MMS capable handsets.

Until recently, mobile capabilities in many parts of the world have been largely limited to voice capabilities or low-speed, data-only sessions. And because services for these networks were originally developed to support voice phone calls, many current pricing models are based around per-minute connect times. For data and multimedia applications, though, connect time pricing models have reduced user enthusiasm for new services. In such a pricing model the slower the network, the more the user pays.

In addition, wireless network operators have begun building higher-speed, packet-switched mobile networks. These networks will better accommodate the new generation of content-rich applications by providing greater bandwidth, eliminating the need to dial network sessions and, in some cases, eliminating the connect-time pricing model. Both 2.5G and 3G networks provide “always-on,” LAN-like connections and this can imply a pay model where you only pay when transmitting or receiving data. This also means that a user can remain connected at no charge and, for example, receive email as it arrives, rather than having to frequently dial up to check for messages.

The combination of richer content, faster networks and the ability to reach a broad set of users with different wireless devices opens up new value-added service opportunities for content providers. In addition, network operators worldwide that have spent billions of dollars on wireless spectrum for delivering next-generation services need to quickly begin earning revenues on their investments. So they are seeking new value-added content and application services to deliver to their own end-customers with mobile and wired Internet access.

There is no doubting the success of the SMS. By August 2000, nine billion SMS messages were being sent each month, including 6 billion in Europe alone. This growth is predicted by Mobile Lifestreams to continue growing until 2004 at least since the mobile phones, infrastructure, specifications, market development and awareness are in place today. Over time, as users connect to networks that offer more advanced data services such as the GPRS and buy mobile terminals that support them, they will use these new bearers for new and existing applications. The new mobile terminals and network will also make it possible to evolve the SMS to a more rich service called MMS. This evolution is natural since there is a big user group for SMS that probably will make use of similar but more advanced services. From the operator's point of view, this evolution is attractive since it will make use of existing user groups and hopefully entice new users, and also create a greater base for profit.

To enhance messaging to this new level, a standard is required. The WAP Forum and the 3GPP 3G industry groups are responsible for standardizing MMS. The MMS standard is created so that the messages may comprise many different network types. To enable messaging in both 2G and 3G the Internet Protocol shall be used as basis of connectivity between these different networks. The standard defines an MMS architecture, which has a number of key elements that interact with each other. MMS will require not only new network infrastructure but also new MMS compliant terminals since MMS will not be compatible with old terminals. This means that before it can be widely used, MMS terminals must reach a certain penetration.

6.2 The prototype

The second issue for the thesis was to build a prototype application for the service presented in chapter 2, using a suitable streaming video platform. An MMS prototype has been developed using a streaming video platform from PacketVideo. The prototype originates from the MMS standard, but some simplifications have been necessary.

The prototype developed should be representative for a service on a future 3G-cell phone. Since 3G cell phones are not available currently, a Compaq iPAQ was used instead. The prototype is separated into one sender agent and one receiver agent.

The sender agent is equipped with a video camera. The video signal from this camera is compressed and encoded using software from PacketVideo, which creates an MPEG-4 file. This file is transferred to a MMS server where it is stored. When the file is completely uploaded, an email is sent to the receiver.

The receiver agent is continuously checking for new “mVideo-email”. When an mVideo-email has arrived, the mVideo-application makes use of information within the email and displays a notification for the user. The notification informs the receiver that a new mVideo message has arrived and asks the user whether he/she wants to see it. If the user decides to see the message, a player is launched with a link to the video file at the server. This starts the streaming video session and the message is displayed on the mobile device.

It was possible to build a prototype of MMS using available technology. However, since 3G cell phones are not available the prototype has not been tested in a 3G network. Instead the prototype has made use of a Nokia Card phone that was plugged into the iPAQ. This proved to work well and it did not prove to be a problem to stream video over a GSM channel at about 30 kbps.

An important conclusion of the prototype is that the platform used from PacketVideo forced us to encode and compress the video signal on a computer running Windows. This meant that the sender part consisted of one iPAQ (acting as a remote control), one laptop and one video camera. In a real situation the camera and the encoder should be integrated in the cell phone.

6.3 Theoretical parameters

The third issue for the thesis was to present problems and technical solutions that are associated with video streaming. It proved that the parameters that affected the video quality could be divided into two categories, parameters connected to the streaming channel, and parameters when encoding/decoding the video signal.

One of the corner stones when communicating over the Internet is the TCP/IP reference model. The TCP/IP model is a “best-effort” service. That means that it does not make any guarantees of delays to applications. This could be a problem when it comes to streaming multimedia applications. There are three major problems for multimedia applications when sending data over the Internet. These are packet loss, end-to-end delay and delay jitter.

Packet loss results in some packets never reaching their destination. This loss of information could be improved by using retransmission mechanisms, Forward Error Correction or Interleaving. The end-to-end delay is the accumulation of transmission processing and queuing delays in routers, propagation delays, and end-system processing delays along a path from source to destination. A delay may result in packets arriving in the wrong order. This is called delay jitter. Delay jitter is not acceptable for a multimedia application, but adding sequence numbers to the packets could help this problem.

To deal with these problems new protocols have been developed by the Internet Engineering Task Force (IETF). These are: Real Time Streaming Protocol (RTSP), Real Time Transport Protocol (RTP) and Real Time Control Protocol (RTCP). These protocols provide functions for monitoring packet losses and re-ordering packets according to sequence numbers. They also implement a sort of “remote control” and add functions such as pause/resume, fast forward and other features often wanted when using multimedia. The RTSP make use of TCP, RTP and RTCP uses UDP.

When it comes to the streaming, there are several techniques to choose from. One is to send a file over UDP at a constant rate equal to the drain rate at the receiver. A second variant is to make use of a buffer and let the client have a few seconds of playout delay. Another way of doing it could be to use TCP and a buffer (in order to avoid packet loss). Finally, there is a possibility to use the scalability option in the MPEG-4 standard and deliver UDP packets with variable data rate. The technique that is chosen has a large impact on the behavior of streamed material, e.g. a buffer decreases the risk of network congestion for the client.

Since the channel to stream through has a number of properties affecting the quality, it is important to choose a compression scheme that takes these into account. The quality of the source is off course vital; a bad source (e.g. video signal) sets an upper limit for quality.

In MPEG-4 the audio and video components are known as objects and one video object may consist of one or more layers to support scalable coding. The MPEG-4 standard can be separated into five profiles that support different features suitable in different situations. An important feature in the MPEG-4 profile Scalable Visual Profile is the rate control. With the rate control, bit streams with both variable rate and constant rate can be used. The scalability makes use of multiple layer features in MPEG-4. PacketVideo uses this technique to enable the delivery of video over wireless channels with highly variable bandwidth. The MPEG-4 standard is also designed for low bit-rate communications devices, which are usually wireless. The MPEG-4 visual standard has been explicitly optimized for three bit ranges: below 64 kbps, 64-384 kbps, and 384 kbps - 4 Mbps.

The MPEG-4 scheme uses different kinds of frames, called I-, P-, and B-Frames. The I-Frames are simply compressed still pictures and use DCT (Discrete Cosine Transform) to compress a single frame without reference to any other frame in the sequence. P-Frames are coded as difference from the last I- or P-Frame. P-Frames make use of motion compensation and DCT coding. As a result P-Frames will give a compression ratio better than I-Frames but depending on the amount of motion present. B-Frames are coded as differences from last or next I- or P-Frame and use both motion compensation and DCT coding. B-Frames use prediction as for P-Frames but for each block either the previous or the next I- or P-Frame. This gives improved compression compared with P-Frames, because it is possible to choose for every macroblock whether the previous or next frame is used for prediction. The use of P- and B-Frames makes it possible to improve compression rate, which is desirable when streaming.

The I-Frames need to be periodically distributed. If all frames depend on their predecessors any end-receiver that misses a frame (e.g. due to packetloss) could never decode any subsequent frames.

6.4 PVPlatform evaluation

The fourth issue for the thesis was to study and evaluate the chosen platform through different channels. This was achieved using an emulation tool making it possible to test how the streaming platform behaves in different kind of network-situations.

It was observed that the I-Frame interval does not seem to have any influence on either the picture quality (as long as no packets are lost during the streaming session), the file size or the network throughput. This depends on the encoder maintaining an almost constant bit rate. When a user tries to put more I-Frames in the bit stream, the encoder will try to maintain the target bit rate by skipping some of the P-Frames and/or reducing the quality of the frames. In the extreme case, if a user tries to encode excessive amounts of I-Frames, all the P-Frames and some I-Frames will be dropped and the amount of data you send through the network will remain approximately equal.

In another test it was observed that the amount of data sent through a 20 MB network differed very much when using the scalability option compared with non-scalable. This probably depends on an optimization issue in the streaming server. There are different QoS settings in the streaming server that have to be based on the underlying network. These QoS settings therefore have an impact on the observed picture quality.

By decreasing the available bandwidth when streaming it was noted that players could get insufficient amount of data in the buffer triggering a network congestion state. From the users point of view this means that the picture is freezing during the network congestion. But it was possible to stream a file encoded for a 512 kbps target network through a network with a bandwidth of only 43.2 kbps without any network error or congestion. It was also observed that it takes about ten seconds before the server starts to adopt for current network conditions and about seven seconds of buffering time before the player starts to play.

The PacketVideo software supports scalable technology and the use of this improved the performance compared to not using it. When a 60 second video sequence encoded with a target bandwidth of 128 kbps was streamed through a 64 kbps network, network congestion was observed during 49 seconds when the scalability was not used, compared to no congestion when using it.

It has also been noted that a high frequency of I-Frames increased the risk of experiencing network congestion. Through discussions with PacketVideo it has been discovered that this phenomena could be explained by the fact that the PVPlatform, depending on the bandwidth and frame rate, is forced to exceed the minimum bit rate. This is due to the fact that each I-Frame will require more bits to encode than a P-Frame, and this constraint may force the targeted minimum kbps to be exceeded. A higher rate of I-Frames will increase the likelihood of network congestion.

When dealing with packet loss the effect of I-Frames was observed. When having a constant 20% packet loss during the whole streaming session, the picture quality grew worse until the appearance of a new I-Frame. By setting the I-Frame rate to 1 second it was observed that the picture quality was improved. A large I-Frame interval implied that the picture quality just grew worse and worse during a long time until the next I-Frame was received.

PacketVideo makes use of the RLP to deal with packet loss and delay jitter. The effect of delay jitter during the tests was very much the same as for packet losses since the PacketVideo platform will not try to reorder the incoming packets.

Acronyms

Acronym	Explanation
3G	Third generation mobile cellular communications systems
3GPP	3rd Generation Partnership Project
ASIC	Application Specific Integrated Circuit
ATM	Asynchronous Transfer Mode
AVI	Audio Video Interleaved
BDP	Bandwidth-Delay Product
BIFS	Binary Format for Scenes
BMP	BitMaP
CDMA	Code Division Multiple Access
CODEC	Compressor/Decompressor
DCT	Discrete Cosine Transform
DMM	Downloading MultiMedia
DMUX	DeMUltipleXor
EMS	Enhanced Messaging Service
GPRS	General Packet Radio Services
GSM	Global System for Mobile communications
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
IP	Internet Protocol
ISO	The International Organization for Standardization
JPEG	Joint Photographic Experts Group
KBPS	KiloBits Per Second
LAN	Local Area Network
MIME	Multipurpose Internet Mail Extension
MMS	Multimedia Messaging Service
MPEG	Motion Picture Expert Group
MUX	MultipleXor
NDA	Non Disclosure Agreement
OS	Operating System
QoS	Quality Of Service
ROM	Read Only Memory
RTCP	Realtime Control Protocol
RTP	Realtime Transport Protocol
RTSP	Real-Time Streaming Protocol
SMM	Streaming Multimedia
SMS	Short Message Service
SMTP	Simple Message Transfer Protocol
SNR	Signal to Noise Ratio
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System

WAP	Wireless Application Protocol
WMF	Wireless Multimedia Forum

References

Documents

[1] 3rd Generation Partnership Project - *3G TS 22.140 V3.1.0 (2000-06) Technical Specification Group Services and System Aspects; Service aspects; Multimedia Messaging Service; Stage 1; (Release 1999)*
Available from http://www.3gpp.org/ftp/Specs/2000-12/R1999/22_series/
Visited 2001-04-09

[2] 3rd Generation Partnership Project - *3G TS 23.040 V3.5.0 (2000-07) Technical Specification Group Terminals; Technical realization of the Short Message Service (SMS); (Release 1999)*
Available from http://www.3gpp.org/ftp/Specs/2000-12/R1999/23_series/
Visited 2001-10-30

[3] 3rd Generation Partnership Project - *3G TS 23.140 V3.0.1 (2000-03), Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional description; Stage 2; (Release 1999)*
Available from http://www.3gpp.org/ftp/Specs/2000-12/R1999/23_series/
Visited 2001-04-09

[4] Audio-Video Transport Working Group – *RFC 1889, RTP: A Transport Protocol for Real-Time Applications (January 1996)*
Available from http://sunsite.dk/RFC/rfc/rfc1889.html
Visited 2001-09-19

[5] Casio – *Product information, E-125 – Cassiopeia*
Available from
http://www.casio.com/personalpcs/product.cfm?section=19&market=0&product=3553
Visited 2001-03-22

[6] Cisco Systems Inc, Documentation – *Understanding TCP/IP*
Available from
http://www.cisco.com/univercd/cc/td/doc/product/iaabu/centri4/user/scf4ap1.htm
Visited 2001-09-19

[7] Compaq – *Product information, iPAQ Pocket PC*
Available from http://www.ipaqsoft.com
Visited 2001-03-26

[8] Compaq – *Product information, iPAQ Pocket PC Expansion Pack*
Available from
http://www.compaq.com/products/handhelds/pocketpc/expansion_packs.shtml Visited 2001-09-10

[9] Cunningham, Francis – *An Introduction to Streaming Video (May 2001)*
Available from http://www.cultivate-int.org/issue4/video/
Visited 2001-09-18

[10] Disctronics –*Introduction to the ISO MPEG standards*
Available from http://www.disctronics.co.uk/video/mpeg.htm
Visited 2001-09-23

[11] DSP Group – *Introduction to Motion Estimation and Compensation*
Available from
http://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/motion/me1.html
Visited 2001-09-22

[12] Ebrahimi, T. and Horne, C. - *MPEG-4 Natural Video Coding – An overview*
Available from http://www.cselt.it/leonardo/icjfiles/mpeg-4_si/7-natural_video_paper/7-natural_video_paper.htm
Visited 2001-09-23

[13] Emblaze – *Company Profile*
Available from
http://www.emblaze.com/serve/about_us/company_profile.asp
Visited 2001-03-22

[14] Emblaze – *Product information, A2Plus Chip*
Available from http://www.emblaze.com/serve/products/a2.asp
Visited 2001-03-22

[15] Emblaze – *Product information, A3*
Available from http://www.emblaze.com/serve/products/a3.asp
Visited 2001-03-22

[16] Emblaze – *Product information, Encoder*
Available from http://www.emblaze.com/serve/products/encoder.asp
Visited 2001-03-22

[17] Emblaze – *Product information, Hardware Solutions*
Available from http://www.emblaze.com/serve/products/hardware.asp
Visited 2001-03-22

[18] Emblaze – *Product information, Wireless Media Platform*
Available from
http://www.emblaze.com/serve/products/media_platform.asp
Visited 2001-03-22

[19] Emblaze – *Product information, Wireless Player*
Available from
http://www.emblaze.com/serve/products/emblaze_player.asp
Visited 2001-03-22

[20] Emblaze – *Product information, Server*
Available from http://www.emblaze.com/serve/products/server.asp
Visited 2001-03-22

[21] Ericsson – *Product information, T68*
Available from http://www.ericsson.com/t68/#
Visited 2001-03-22

[22] Ericsson – *Information, third generation*
Available from http://www.ericsson.com.au/3G/3g_business_case.asp
Visited 2001-03-26

[23] Fossa, Carl E. – *An Analysis of Quality of Service Enhanced Internet Protocol in Third-Generation Wireless Networks (April 2000)*
Available from http://fiddle.visc.vt.edu/courses/ecpe6504-wireless/projects_spring2000/report_fossa.pdf
Visited 2001-09-20

[24] Koenen, Rob. *MPEG-4 – Multimedia for our time.*
IEEE Spectrum February 1999 Volume 36 Number 2
Available from: http://www.cselt.it/mpeg/documents/koenen/MPEG-4.htm
Visited 2001-04-05

[25] McConnaughy, Rozalynd. *Streaming Audio*
Available from
http://students.libsci.sc.edu/clis743/GraphicsEditors/rpm_r2.htm
Visited 2001-07-17

[26] Microsoft Windows CE 3.0 Datasheet
Available from
http://www.microsoft.com/windows/embedded/ce/guide/datasheets/ce30datasheet.asp
Visited 2001-09-11

[27] Mobile Lifestreams – *Next Messaging, An Introduction to SMS, EMS and MMS*
Available from http://www.NextMessaging.com
Visited 2001-04-05

[28] Network Working Group – *RFC 2326, Real Time Streaming Protocol (April 1998)*
Available from http://sunsite.dk/RFC/rfc/rfc2326.html
Visited 2001-09-19

[29] Network Working Group – *TCP over 2.5G and 3G Wireless Networks (July 2001)*
Available from
http://www.ietf.org/internet-drafts/draft-ietf-pilc-2.5g3g-03.txt
Visited 2001-09-20

[30] NIST Net Home page
Available from http://snad.ncsl.nist.gov/itg/nistnet/
Visited 2001-07-05

[31] PacketVideo – *Corporate information*
Available from http://www.packetvideo.com/about/
Visited 2001-03-20.

[32] PacketVideo – *Corporate Fact Sheet.*
Available from http://www.packetvideo.com/press/docs/pv_corp.doc
Visited 2001-03-16

[33] PacketVideo – *FrameTrack Performance in Simulated GPRS environment (July 2001)*
Received from PacketVideo, not a public document

[34] PacketVideo – *Multimedia Technology Overview – Standards, Algorithms, and Implementations*
Available from http://www.pv.com/pdf/pv_whitepaper.pdf
Visited 2001-07-16

[35] PacketVideo – *PVPlatform Product Description (2001)*
Received from PacketVideo, not a public document

[36] Internet product watch – *PVPlatform 2.0 (PacketVideo Corp)*
Available from
http://ipw.internet.com/development/wireless/966953978.html
Visited 2001-03-20

[37] PacketVideo – *Product information*
Available from http://www.packetvideo.com/wireless/products.asp
Visited 2001-03-20

[38] PacketVideo – *Wireless Media, technology*
Available from http://www.packetvideo.com/wireless/tech.asp
Visited 2001-03-22

[39] PacketVideo – *Wireless Media – Products*
Available from http://packetvideo.com/prodtech/products.asp
Visited 2001-09-11

[40] Rana, Omer F. – *Multimedia Services*
Available from http://www.cs.cf.ac.uk/User/O.F.Rana/data-comms/comms-lec11.pdf
Visited 2001-09-18

[41] Shuler, Rus – *How does the Internet Work? (1998)*
Available from
http://rus1.home.mindspring.com/whitepapers/internet_whitepaper.html
Visited 2001-09-18

[42] Sweden awards UMTS licenses
http://www.3gnewsroom.com/3g_news/news_0081.shtml
Visited 2001-03-26

[43] Wireless Multimedia Forum – *Delivering Streaming Media to the Mobile Masses, (February 2001)*
Available from http://www.wmmforum.com/public_documents.htm
Visited 2001-08-28

[44] Wireless Multimedia Forum – *Recommended Technical Framework Document, Version 1.0*
Available from http://www.wmmforum.com/public_documents.htm
Visited 2001-08-28

[45] Wireless Multimedia Forum – *The Business Case for Wireless Multimedia Services, (May 2001)*
Available from http://www.wmmforum.com/public_documents.htm
Visited 2001-08-28

[46] Woodward, Peter - *Introduction to MPEG-4 (Draft) (June 2000)*
Available from http://www.dcs.qmul.ac.uk/~petew/html/IntroMP4.html
Visited 2001-09-23

Correspondence

[47] Hebert, Alan – PacketVideo
Email
2001-07-31

[48] Rydén, Jacob – Ericsson
Phone call
2001-09-11, 14:30

[49] Dr. Wang, Szu-Wei – PacketVideo
Email
2001-09-11

Literature

[50] Forchheimer, Robert (1999), *Image Coding and Data Compression*, Dept. of Electrical Engineering, Linköping Universitet

[51] Kurose, J F, Ross, K W (2001), *Computer Networking – A Top-Down Approach Featuring the Internet*, Amherst, ISBN 0-201-47711-4.

[52] Tanenbaum, Andrew S (1996), *Computer Networks – Third Edition*, New Jersey, ISBN 0-13-349945-6.