Michael Chukwu

Adaptive Mobile Multimedia Services

Master's Thesis
in Information Technology
May 24, 2007

University of Jyväskylä

Department of Mathematical Information Technology

Jyväskylä

Author: Michael Chukwu
Contact information: michukwu@cc.jyu.fi
Title: Adaptive Mobile Multimedia Services
Työn nimi: Sopeutuvat/oppivat multimedia mobiilipalvelut
Project: Master's Thesis in Information Technology
Page count: 91
Abstract: The delivery platforms for multimedia content has increased from television and desktop computers to cell phones and portable media players. However transparent utilization of these services requires the ability to take existing multimedia content and change the format, bitrate and/or resolution in order to view it on another playback device.
Suomenkielinen tiivistelmä: Multimediasisällön jakamiskanavat ovat lisääntyneet televisiosta ja tietokoneesta puhelimiin ja kannettaviin mediasoittimiin. Mutta kuitenkin näiden palveluiden näkymätön käyttö vaatii kykyä muuttaa olemassaolevan sisällön formaattia, pakkaustiheyttä ja/tai resoluutiota, jotta sen voi toistaa toisenlaisella mediasoittimella.
Keywords: Multimedia, Wireless, Network
Avainsanat: Multimedia, Langaton, Verkko

Contents

1 Introduction

There is a growing trend in the deployment of multimedia services to mobile platforms ranging from cellular phones to personal digital assistants. These services have consistently grown from predominantly standalone capture and playback applications to broadcast, downloadable and streaming services. Broadcast systems like DVB–H (Digital Video Broadcasting – Handheld) and MBMS (Multimedia Broadcast Multicast Services) provides resource efficient multimedia services to mobile platforms but places effective limit on the available content due to their broadcast nature. This eliminates the possibility for true and full user driven access to multimedia content, especially to the internet. However an unprecedented volume of multimedia services are available on the internet today, there are thousands of live streaming media services, offering news cast, entertainment and sports content. In addition there exists countless and ever increasing number of video and game on–demand services. The access and consumption of these multimedia services has been the exclusive preserve of wireline internet and desktop computers. This is due to high resource requirements in terms of network transmission bandwidth, device processing and display capabilities. Also wireless communication links introduce additional bottlenecks in terms of high losses, errors and delay rates. This is caused by fading, interference, network congestion and user mobility. Moreover, the radio environment dynamics, makes it difficult to provide a guaranteed network service.

However with the recent trends in mobile/wireless technologies, significant increases in data transmission rates have been recorded and more is expected. The most significant of these new trends is the High Speed Downlink Packet Access (HSDPA)[1] and High Speed Uplink Packet Access (HSUPA) technologies of the UMTS and the revision 'e' of IEEE 802.16 – Wireless Metropolitan Access Network (WMAN) [2] also known as Mobile WiMax. With these recent additions to the community of mobile/wireless technologies, there exists a credible potential of duplicating comparable data transmission speeds of wireline networks on mobile/wireless networks. More so, the overall information processing and display capabilities of mobile devices has recorded a giant leap, flaunting more than adequate processing power, memory and display capabilities for multimedia services.

Nevertheless mobile access to internet (and/or network) multimedia services is still limited, this limitation is due to problems of incompatibility. There exists plethora of multimedia formats, and these have increased and diverged as stakeholders and technologies multiply. In addition, the capabilities of mobile devices vary widely in terms of processor speed, display constraints, network connectivity, power constraints and audio/video decoding capabilities. This presents a huge constraint to the transparent utilization of multimedia services on mobile platforms. This scenario has resulted in the provision of multimedia content for specific target devices and network connectivity environment by service operators and content providers. The burden of providing multimedia data streams that correctly (let alone optimally) satisfy all the variable device/user characteristics, places a limit on the available content to mobile users.

However, several researches tend to address this phenomenon; prominent among the efforts in solving this problem is the MPEG 21 â- Universal Multimedia Access [3]. This is a very laudable standard with an all encompassing solution to the problem of diversity between multimedia services and usage environment. However, strict implementation of this framework excludes all other competing multimedia formats which form a very significant size of the available internet multimedia services.

In order to provide an effective connection between internet multimedia services and mobile delivery platforms and support complete transparency in accessing these multimedia services on mobile platforms, a content adaptation solution, provided as a multimedia proxy application is proposed. The proxy provides the essential functions of converting the multimedia streams to the device specific formats, adapting to the error and delay properties of devices' network access technologies and real – time responses to congestion of transmission channels thereby maintaining overall Quality of Service (QoS) that leads to an acceptable user experience.

1.1 Background

Deployment of multimedia services to mobile communications networks has dominated several recent research efforts, the focus of these researches can be broadly classified into three(3) major functional requirements for mobile multimedia services. These includes : network infrastructure, communications protocols and application design requirements.

Network infrastructure plays a crucial role in multimedia communications as it provides support for traffic bandwidth requirements, sustains the interactivity of multimedia systems and supports the higher layers of communications protocols and applications design requirements. This group of research[4, 5, 6, 7, 8, 9, 10, 11, 12, 13] focuses on the design and deployment of communication infrastructure that supports shared and real–time applications with wide ranges of bandwidth and latency requirements.

Both wireline and wireless media were considered in their research and the salient characteristics that could provide the required support for multimedia communication were highlighted. The twin concepts of interconnectivity and interoperability were also explored with the need to maintain interactivity and seamless inter–operation across wide communication infrastructures of different compositions underscored. In addition, reliabilty and error–correction for different media types were critically examined with proposals of novel solutions and techniques for enhancements.

In addition to infrastructure is the communication protocol requirements; communication protocols provide routing, handovers, packet fragmentation/re–assembly, flow control and error correction functions. The impact of these functionalities on multimedia communications have been the subject of interest in various research communities, yielding various techniques, implementations and standardizations. Several researches [14, 15, 16, 17, 18, 19, 20, 21, 22] appraised the legacy transport protocols available in TCP/IP suite namely UDP (user datagram protocol) and TCP (transmission control protocol), with all agreeing on the inadequacy of the later (TCP) in multimedia communications by presenting features considered inimical to this objective which included bursty transmissions, frequent fluctuation of transmission rates due to congestion control and delay insensitive error control mechanisms. However a feature termed TCP–friendliness which represents fair allocation of transmission bandwidth among competing data traffic was isolated as good and important. However UDP transport protocol was presented as supportive of multimedia communications but lacking in key issues of error corrections and flow control. Several propositions and implementations for the augmentation of both transport protocols were presented and the Real–time Transport Protocol (RTP) presented later in this thesis is one example of this effort.

Lastly the subject of network multimedia application design is heavily researched with various methods and solutions proposed, simulated and implemented. This group [23, 24, 25, 26, 27, 28] of research tends to focus on adaptation of multimedia content for internet delivery. In their formulation of the research problem, they presented the constraint of viewing multimedia rich web pages on devices and personal computers of different access speeds and resources characteristics. While another group of research [29, 30] focuses largely on multimedia adaptation for delivery across a variety of users and terminals. Their research problem range from adapting pre–stored, live and/or third party pre–encoded multimedia content to variety of users and terminals. More so the MPEG–21 Universal Multimedia Access [31]; presents an all encompassing effort in the research, design and implementation of a multimedia adaptation system. The concepts expressed by these researches form a huge and growing body of knowledge and a strong base for this research.

1.2 Research Problem

Providing transparent access to internet and related network multimedia services on mobile platforms requires solution to the following identified problems :

1.) Incompatibility of properties/formats of these multimedia services and that of the mobile devices. The wide variety of multimedia formats encoded with varying temporal, spatial and rate properties makes it difficult to utilize these on mobile platforms.
2.) Diversity in users' service levels and access technology types. This requires adapting to the several data rates of each service level and also to the data transmission and error properties of mobile/wireless access technologies.

1.3 Thesis Layout

The thesis is presented in seven(7) chapters. The first chapter – the introduction, presents a brief but concise information on the research problem and the background to the current effort. The second chapter provides review of all related multimedia systems and standards. The various evolutions of these standards to the current status are highlighted. The third chapter reviews all network related multimedia systems and applications. The standards, technologies, trends and future of these applications and systems. The fourth chapter – the system design proposes a framework for real–time delivery of multimedia services to user terminals across mobile/wireless networks. The fifth chapter – implementation contains the technical description and implementation details of the proposed system. The sixth chapter provides the description and analysis of the test bed, parameters and results. The last and seventh chapter presents the conclusion on the achievements of the systems against the set objectives. The non–chapter sections of the references and appendix presents details of the referred materials and other relevant information.

2 Multimedia Systems Review

Multimedia refers to a collection of two or more of the following media types : text, graphics, images, audio and video. The media can be grouped into two classes : static (text, graphics and images) and dynamic (animations, audio and video). The dynamic media can be further classified into non–continuous (animations) and continuous (audio and video) media. These classifications are based on the temporal properties of the data content.

Multimedia systems confronts a specific problem of handling large volumes of data; representation of multimedia information requires large amounts of data, and this tends to increase significantly along the path of the classification chain. For example, the size of one page of text encoded with UTF–8 (UTF–8 requires between one to four bytes per character) at eighty characters per line and 64 lines per page will be 20.4 kBytes. In same vein, the size of a bitmap image with resolution of 704 x 576 is 1.2 MBytes. However for continuous multimedia data, for example CD – quality audio streams at sampling rate of 44.1 kHz, stereo and sample size of 16 bits will generate 1.4Mbits of data per second. Along the same path of classification, an AVI (Audio Video Interleave – a video encapsulation format) video containing bit–map images of 4CIF (704 x 576) resolution at 30 fps will result to 292 Mbits of data per second.

Following the exceptionally large amounts of data generated by continuous multimedia systems. The storage requirements would be extremely expensive to satisfy. However, besides the storage requirements, rendering video or audio in real time would be very difficult due to incompatible access speeds of current storage devices. To play back the earlier described AVI video would require access speed of 292 Mbits/s (36.5MBytes/s). However, today's CD–ROM technology provides a transfer rate of about 7.8MBytes/s (52x speed). More so, transmission of multimedia data over a communication network would be totally inconceivable. At the present state of storage devices and communications technologies, the only solution is to compress the data before storage/transmission and decompress it before playback.

Compression techniques play a crucial role in digital multimedia applications and are required for three reasons: large storage requirements of multimedia data, relatively slow storage devices that cannot play multimedia data (specifically video) in real time, and network bandwidth that does not allow real–time video data transmission [32].

Computational algorithms for multimedia data compression have been developed in response to this challenge. The techniques employed in these algorithms are classified into lossless and lossy techniques. In lossless techniques, the compression and decompression process occurs without any losses as the original data is fully recoverable, while the original data cannot be recovered fully in lossy techniques.

Lossy techniques supports higher compression ratios in comparison to their lossless counterparts, however combinations of these two techniques are most used, especially for video compression. Compression algorithms employ several techniques for sampling and transforming of data prior to compression for easier processing. These includes prediction–, frequency–, and importance–based techniques. Predictive techniques as the name implies, predicts the current values from the previous values and thereafter encodes the difference between the predicted values and the actual values. Frequency–oriented techniques (specifically for images) transforms the images from the spatial domain to the frequency domain, the commonly used transformation techniques are the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and Fast Fourier Transform (FFT). Importance–oriented techniques use other characteristics of media as the basis for compression.

Video systems generates the largest amounts of data in comparison to other media systems, this is responsible for the comparative complexity of video compression schemes. The basic design of a digital video compression algorithm is structured to expliot the redundant nature of video images. Video content is made up of series of still images otherwise called frames, these frames are organized in order of their temporal relationship, hence maintains close proximity in content. The changes across frames of a given sequence is very minimal, if existent. In addition, redundancies exist in the individual frames; for example some pixel areas could have the same colour. The compression algorithms removes spatial redundancy within a video frame and temporal redundancy between video frames. Signal transformation (this could be DWT, DCT or FFT) and quantization is used to reduce spatial redundancy, while motion–compensation is used to exploit temporal redundancy between frames. Motion–compensation technique is used to encode a video frame based on its relationship with other video frames temporally close to it. The general equation for 8x8 2D image DCT is presented below:

$$F(u,v) = \frac{c(u)c(v)}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} cos\left[\frac{\pi u}{16}(2i+1)\right] cos\left[\frac{\pi v}{16}(2j+1)\right] f(i,j)$$

where u, v = 0, 1, ... , 7
if u, v = 0 then c(u), c(v) = $\frac{1}{\sqrt{2}}$
if otherwise, c(u), c(v) = 1

In a typical video compression process, video frames can be acquired in a digitized standard RGB (Red Green Blue colour representation) format, with each picture element(pixel) represented by a 24–bits value (8 bits each for Red, Green, and Blue), and thereafter is converted to YUV (Y – brightness value, U – colour difference to blue and V – colour difference to red) format.

In YUV format, images are also represented in 24 bits per pixel (8 bits for the luminance (brightness) information (Y) and 8 bits each for the two chrominance (colour difference) information (U and V)). Thereafter the YUV formatted data is subsampled. In this process all luminance information is retained. However, chrominance information is subsampled 2 : 1 in both the horizontal and vertical directions. This results to 2 bits per pixel of U and V information. This subsampling does not drastically affect quality because the eye is more sensitive to luminance than to chrominance information. Subsampling is a lossy step. The 24 bits RGB information is reduced to 12 bits YUV information. Thereafter the frames are divided into 16x16 pixel macroblocks. Each macroblock consists of four 8x8 luminance blocks and two 8x8 chrominance blocks (1 U and 1 V). Macroblocks are the units used for motion–compensated prediction while the blocks are used for signal transformation. the resulting data from signal transformation is quantized, this is used to constrain the size of data generated and thereby control the bitrate. The data is quantized according importance as illustrated in Figure 2.3, in this scenario the resultant 8x8 array (Fig. 2.2), is divided by a non–uniform 8x8 array of weighted values that corresponds to importance level of the pixel values.

The resulting data is zig–zag ordered (Fig. 2.4), this is the transformation of the 8x8 array into 1x64 linear array; and the linear array is run–length encoded, this removes redundancies and reduces the array to a more compact form. Finally the data is entropy encoded.

The frames can be encoded in three types: intra–frames (I–frames), forward predicted frames (P–frames), and bi–directional predicted frames (B–frames). An I–frame is encoded as a single image, with no reference to any past or future frames. A P–frame is encoded relative to the past P– or I–frame and a B–frame is encoded relative to the past, future, or both P– or I– frames. The encoding for B–frames is similar to P–frames, except that the B–frame are not used for reference.

231	224	224	217	217	203	189	196
210	217	203	189	203	224	217	224
196	217	210	224	203	203	196	189
210	203	196	203	182	203	182	189
203	224	203	217	196	175	154	140
182	189	168	161	154	126	119	112
175	154	126	105	140	105	119	84
154	98	105	98	105	63	112	84

Figure 2.1: The original image data

174	19	0	3	1	0	-3	1
52	-13	-3	-4	-4	-4	5	-8
-18	-4	8	3	3	2	0	9
5	12	-4	0	0	-5	-1	0
1	2	-2	-1	4	4	2	0
-1	2	1	3	0	0	1	1
-2	5	-5	-5	3	2	-1	-1
3	5	-7	0	0	0	-4	0

Figure 2.2: The resulting 8x8 matrix after a 2D image DCT [33]

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

Figure 2.3: The 8x8 Quantization matrix [33]

174	19	0	1	0	0	0	0
52	-13	-2	-1	-1	0	0	0
-9	-2	4	1	0	0	0	0
1	3	-1	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 2.4: Quantized Matrix of DCT values

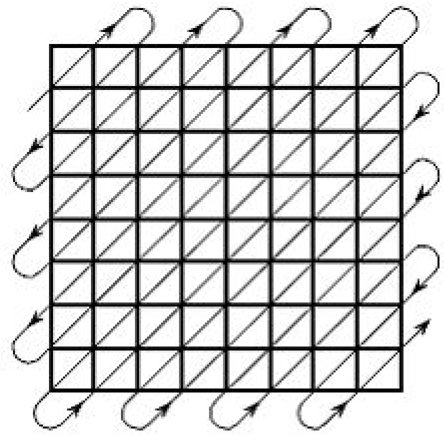

Figure 2.5: The ordering of the quantized DCT coefficients in a zig–zag sequence

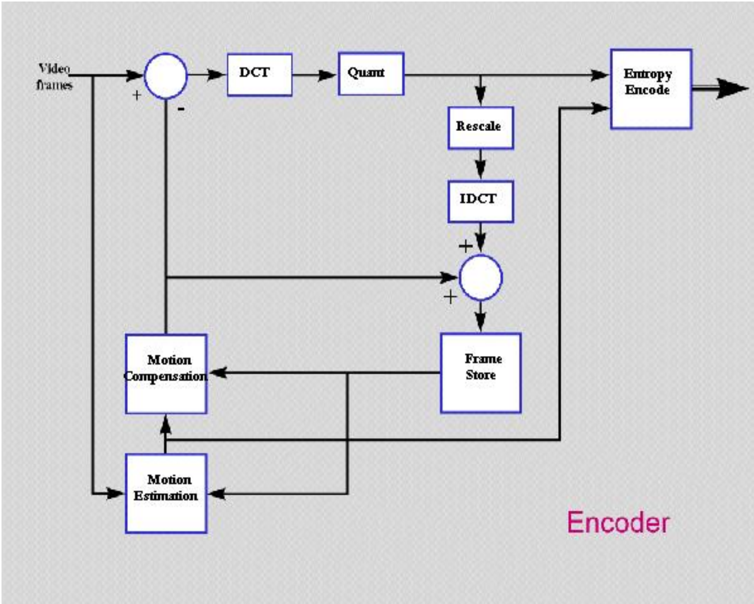

Figure 2.6: General Structure of a Video encoder using DCT signal transformation based on motion compensated prediction[34]

2.1 Multimedia Standards

Various groups have established standards for digital multimedia compression based on a particular application (and/or modification) of existing techniques. These standards combine several algorithms and approaches, including lossy and lossless schemes to form an optimized set of procedures for digital multimedia compression. The H.261 standard is the first ever digital coding standard, authored by the International Telecommunication Union, the arm of the United Nations responsible for telecommunication standards. This was a significant landmark in digital video compression, designed for the transmission of digital video on ISDN networks. Following the H.261 was MPEG–1 developed by the Motion Pictures Expert Group and the H.262 jointly developed and published by ITU–T and MPEG (MPEG–2 standard). The H.263 was the next video compression standard developed, as an improvement in the earlier designs of H.261, MPEG–1 and MPEG–2 with specific targets of low bit–rate encoding for videoconferencing, on both circuit– and packet–switched networks. Several amendments to the standard has been released with its latest version H.263++ (or H.263v3) standardized in 2000, this has resulted in improved quality, compression efficiency and error resilience. The MPEG–4 and the H.264 video coding standards are the lastest additions from these organisations.

This section is a presentation of the prominent standards amongst the plethora of formats available.

2.1.1 MPEG–1

MPEG–1 was the first standard developed by the Moving Picture Experts Group (MPEG) of International Standards Organization/International Electrotechnical Commission (ISO/IEC). MPEG–1 improved upon the H.261 motion–compensated prediction scheme by integrating field interpolation for the video part and by developing a cost–effective sub–band coding scheme for the audio part. This brought virtual transparency of audio at a low bit rate of 128kbit/s per audio audio channel and sampling rate of 48kHz.

2.1.2 MPEG–2

MPEG–2 extended the basic MPEG–1 system to provide support for TV quality transmission of digital video. The MPEG–2 project started July 1990 meeting in Porto, Portugal [36] to provide a solution that in four years led to a standard for digital television transmission. The compression algorithm [37] provides a coding standard to efficiently encode interlaced video, and multi–channel audio [37] that preserves backward compatibility with MPEG–1. It uses the same techniques as MPEG–1 but defines two methods of combining of one or more elementary streams of video and audio, as well as, other data into single or multiple streams which are suitable for storage or transmission. The storage format is defined as the Program Stream and the transmission format is defined as the Transport Stream.
The Program Stream combines one or more Packetised Elementary Streams (PES), which have a common time base, into a single stream. The PES consists of one type of data (video or audio). The Program Stream is designed for use in relatively error–

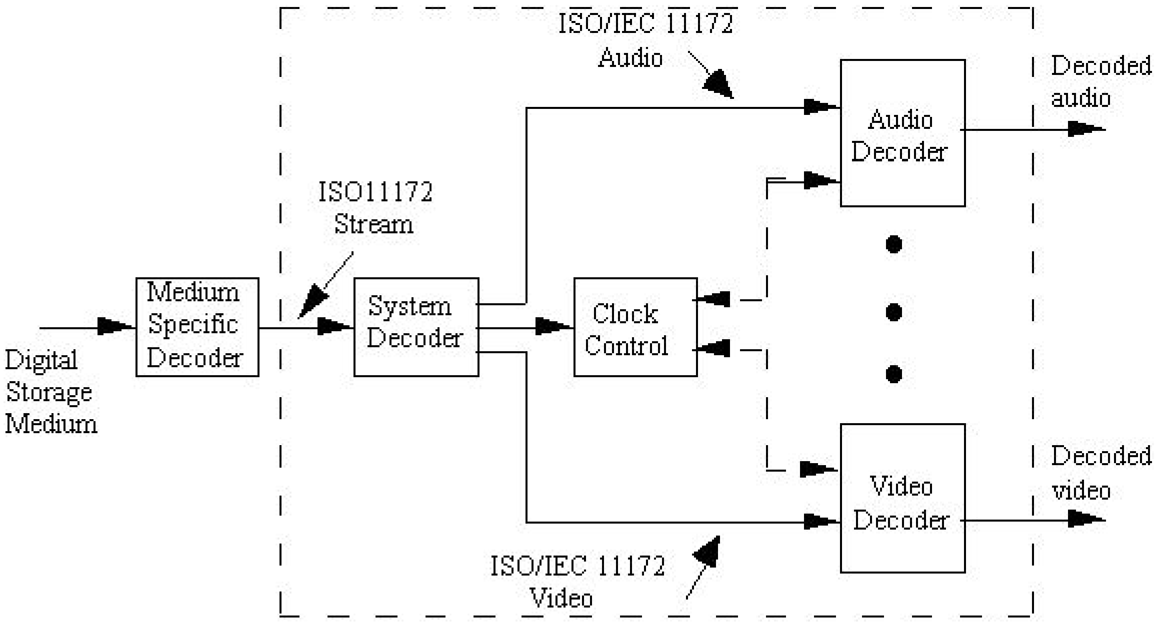

Figure 2.7: Prototypical MPEG–1 (ISO/IEC 11172) decoder [35]

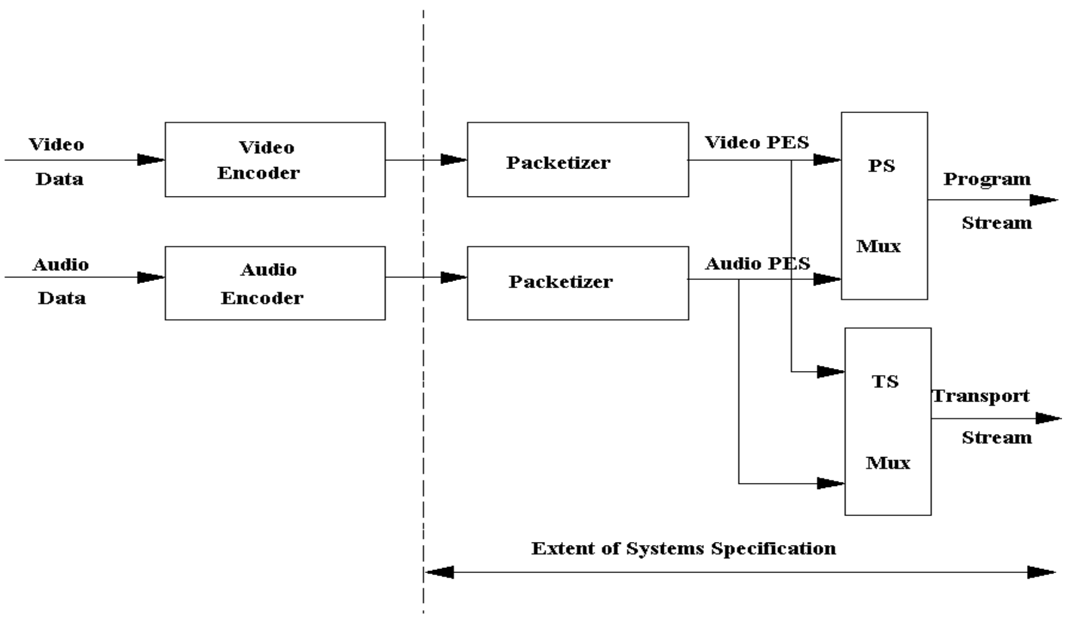

Figure 2.8: Model for MPEG–2 Systems [37]

free environments. Program stream packets may be of variable length. The Transport Stream also combines one or more Packetized Elementary Streams with one or more independent time bases into a single stream. Elementary streams sharing a common timebase form a program. The Transport Stream is designed for use in environments where errors are likely, such as storage or transmission in lossy or noisy media.

MPEG defines a set of tools called profiles [38, 39], they define a set of interoperability points for implementations from different parties. The profiles are further divided into levels. The levels limit complexity of tools used within that defined grouping.

MPEG–2 is defined in nine [37] parts. A brief description of some of the parts is presented below:
Part 1 of MPEG–2 specifies the format for combining of one or more elementary streams of video, audio, and/or data into a single or multiple streams for storage or transmission.
Part 2 specifies the resolutions and qualities known as profiles and levels, the coding structure offers a wide range of coding tools for different functionalities, which includes scalable video coding, a technique for encoding video in different levels, with a basic definition and additional enhancements known as base and enhancement layers respectively. Four scalability factors are defined and supported in two profiles
Part 3 is the backward–compatible multichannel extension of the MPEG–1 Audio standard.
Part 6 The Digital Storage Media Command and Control (DSM–CC) is the specification of a set of protocols which provides the control functions and operations. These protocols offer ability to use a return channel to control the content and scheduling of the bitstreams. The DSM–CC is designed in a client and server structure to support managed interaction of the users and bitstreams' content
Part 9 specifies the Real–time Interface (RTI) to Transport Stream decoders, which may be utilised for adaptating the stream to all appropriate networks carrying the transport streams.

2.1.3 MPEG–4

Low bit–rate multimedia streaming over IP networks was the next natural demand from the market after the MPEG–2 success with DVD and satellite digital broadcasting [40]. The original goal was to make a standard for low bit rate applications, as earlier standards already covered high rate applications, but at the specification phase,

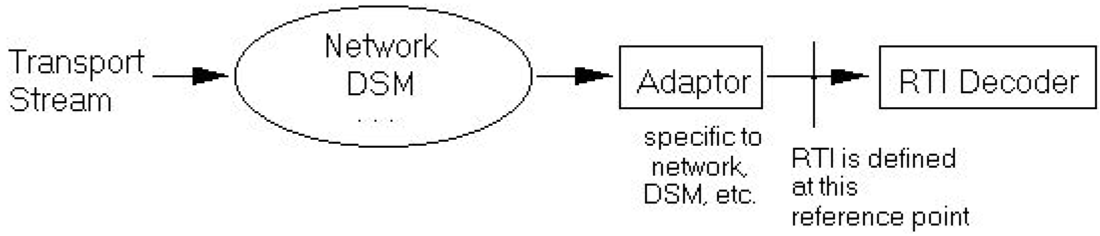

Figure 2.9: Reference configuration for the Real-Time Interface [37]

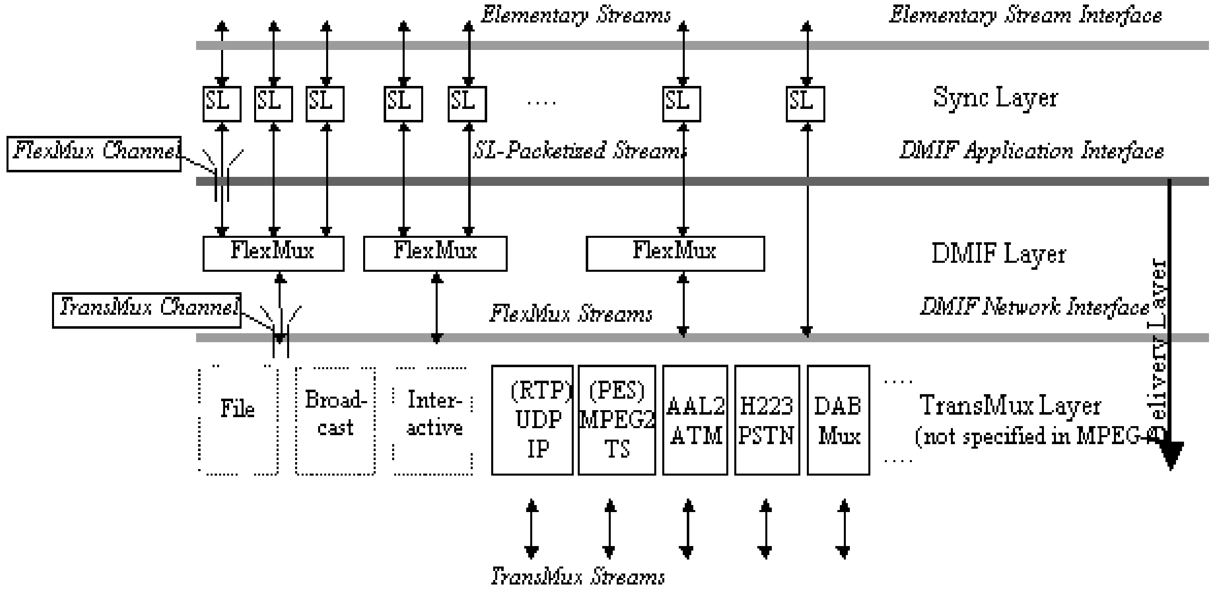

Figure 2.10: The MPEG-4 System Layer Model [34]

MPEG–4 was expanded into a standard dealing with high compression ratios covering both low and high bitrates[38].

It contains a set of specifications[34] for:

1) Representing aural, visual and audiovisual content called media objects. These objects can be natural or synthetic.
2) Delivering of these independent layered media objects over heterogeneous networks by using streaming protocols.
3) Rendering and presenting of scenes dynamically.
4) Providing a platform for the receiver's interaction with the audiovisual scene generated at the decoder phase.

There are several parties [40, 41] involved in the development and implementation of MPEG–4, they include: 3GPP, ISMA, IETF, M4IF and WMF. The 3GPP (The 3rd Generation Partnership Project), is a global cooperation between six organizational partners (ARIB, CWTS, ETSI, T1, TTA and TTC); the world's major standardization bodies from Asia, Europe and USA. 3GPP produces technical specifications and reports for 3rd Generation Mobile System based on evolved GSM core networks and the radio access technologies that they support. The Internet Streaming Media Alliance (ISMA) is a consortium of companies developing specifications and products that use parts of MPEG–4 as well as non–standard extensions and Internet Engineering Task Force (IETF) is an international community of network designers, operators, vendors and researchers. They develop audio–video payloads for the Real–time Transport Protocol just to name a few of their activities. M4IF's (MPEG–4 Industry Forum) focus is to facilitate the adoption of MPEG–4 standard by establishing MPEG–4 as an accepted and widely used standard among application developers, service providers, content creators and users. Whereas Wireless Multimedia Forum (WMF) aims at establishing technology consensus around a set of protocols that are suitable for use in streaming multimedia over a wireless network.

2.1.4 MPEG–7

An incommensurable amount of audiovisual information is becoming available in digital form, in digital archives, on the World Wide Web, in broadcast data streams and in personal and professional databases, and this amount is only growing. However the value of information often depends on how easy it can be found, retrieved, accessed, filtered and managed.

"Multimedia Content Description Interface," also known as MPEG–7, is a standard designed to provide content that is associated with a sets of descriptors [42, 43]. These provide a great deal of information about the Audio Visual (AV) content for fast searching and retrieval by a variety of tools including search engines, databases, and other multimedia applications.

MPEG–7 addresses applications that can be stored (on–line or off–line) or streamed (e.g. broadcast, push models on the Internet), and can operate in both real–time and non real–time environments. A 'real–time environment' in this context means that the description is generated while the content is being captured.

The simple and highly abstract multimedia annotation consumption cycle includes: feature extraction (description generation), the description itself, and the usage application.

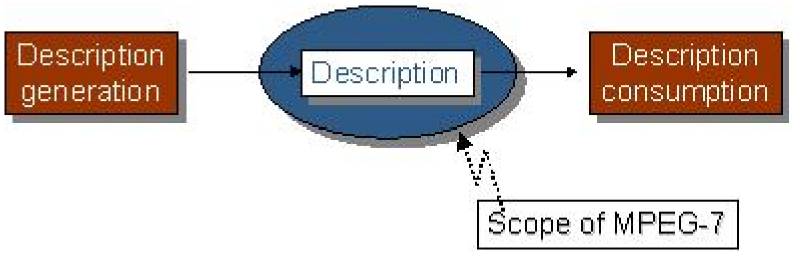

Figure 2.11: The Overview of the MPEG–7 standard : The scope of the standard is strictly on the content description [43]

From a multimedia content an audiovisual description can be obtained via manual or automatic extraction. The generated AV description may subsequently be stored or streamed directly. Client applications will submit queries to the descriptions repository if stored and will receive a set of descriptions matching the query for browsing (just for inspecting the description), for manipulating it and/or for retrieving the described content, etc. In a push scenario a filter (e.g., an intelligent agent) will select descriptions from the available ones and perform the programmed actions afterwards (e.g., switching a broadcast channel or recording the described stream).

MPEG–7 is aware of, and took into account, the activities of a number of other standards groups during the development process [44]. For the archival descriptions, library (e.g., MARC, Z39.50) and archive (e.g., EBU/SMPTE, ISAD (G), EAD, Dublin Core, CEN/ISSS MMI) standards was taken into account. While for the streaming descriptions, the broadcast Electronic Programme Guides (EPGs) (e.g., DVB, ATSC) and web channels (Channel Definition Format (CDF)) standards was considered. For the intellectual property and rights management descriptions, a liaison was formed with the INDECS project. The DDL group of MPEG–7 has been closely monitoring the work of the W3C's XML Schema Working Group and the XLink, XPath and XPointer Working Groups.

The MPEG–7 community is attempting to combine efforts with these groups through liaisons. This will hopefully maximize interoperability, prevent duplication of work and take advantage of work already done through the use of shared common ontologies, description schemes and languages. MPEG–7 hopes to act as a gateway or container for older established standards while at the same time providing a reference standard which can be used by proprietary multimedia applications or specific multimedia domains.

MPEG–7 Parts:	**Descripton**
MPEG–7 Systems	The tools needed to prepare MPEG–7 descriptions for efficient transport and storage and the terminal architecture.
MPEG–7 Description Definition Language	The language for defining the syntax of the MPEG–7 Description Tools and for defining new Description Schemes
MPEG–7 Visual	The Description Tools dealing with (only) Visual descriptions.
MPEG–7 Audio	The Description Tools dealing with (only) Audio descriptions.
MPEG–7 Multimedia Description Schemes	MPEG–7 Multimedia Description Schemes (also called MDS) comprises the set of Description Tools (Descriptors and Description Schemes) dealing with generic as well as multimedia entities.
MPEG–7 Reference Software: the eXperimentation Model	The eXperimentation Model (XM) software is the simulation platform for the MPEG–7 Descriptors (Ds), Description Schemes (DSs), Coding Schemes (CSs), and Description Definition Language (DDL).
MPEG–7 Conformance	MPEG–7 Conformance includes the guidelines and procedures for testing conformance of MPEG–7 implementations.
MPEG–7 Extraction and use of descriptions	Informative material about the extraction and use of the Description Tools
MPEG–7 Profiles	MPEG–7 Profiles and levels – provides guidelines and standard profiles.
MPEG–7 Schema Definition	This specifies the schema using the Description Definition Language

Table 2.1: Brief description of the major functionalities offered by different parts of MPEG–7 [43]

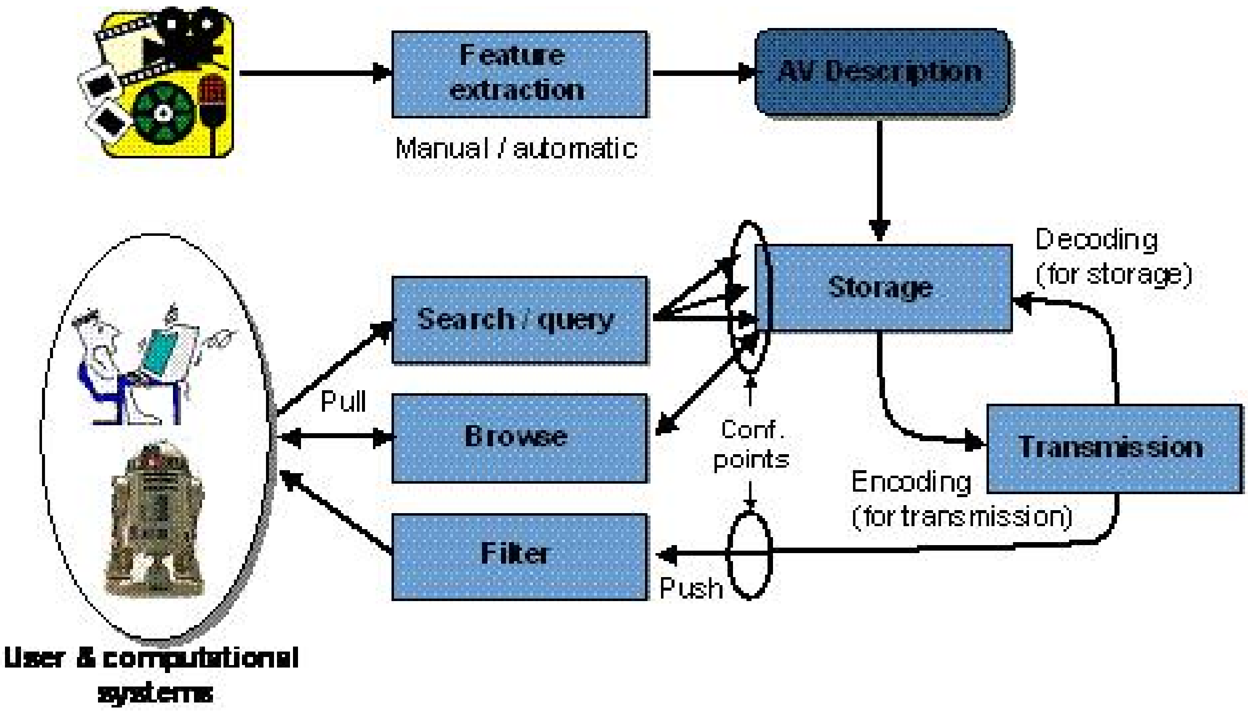

Figure 2.12: The MPEG–7 standard: Feature extraction and content description scenario [43]

MPEG–7 application areas include Education, Journalism, Entertainment, Tourism, Medical Applications, Archives (Film, Video, and Audio), Shopping and Social.

2.1.5 MPEG–21

MPEG–21 titled 'Universal Multimedia Access' is a standard that defines a framework for collaborative and transparent utilization of distributed multimedia services. Unlike the previous standards, MPEG–21 defines a standard that covers the entire life cycle of multimedia services starting from creation and distribution to delivery and consumption[31].

MPEG–21 Multimedia Framework Part 1 [31, 45]: titled "Vision, Technologies and Strategy" states the fundamental purpose of the framework as to:

Define a 'vision' for a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices to meet the needs of all users, achieve the integration of components and standards, to facilitate harmonisation of 'technologies' for the creation, management, transport, manipulation, distribution and consumption of digital items.

Define a 'strategy' for achieving a multimedia framework by the development of specifications and standards based on well–defined functional requirements through collaboration with other bodies.

MPEG–21 [3] is designed to be a baseline and a standard for co–operative design and implementation of a complete solution. Two aspects of interoperable multimedia communication provided in a three–tier architecture are described: the consumer side of media delivery; and the provider angle of the multimedia content generation and adaptation. Adaptation is the third component of the three tier architecture but its design and implementation is not part of the MPEG–21 design.

Content Consumer: The MPEG–21 framework has a central objective of providing the ability to play/consume (multimedia) content independently of the presentation device. In pursuant to this objective, the framework has formulated an interoperable description format of the device's capabilities. The usage environment description (UED), part of the MPEG–21 Digital Item Adaptation (DIA) specifies how to describe such devices in terms of their codec capabilities, I/O capabilities, and device properties.

Furthermore, the UED defines description formats for the networks through which the contents are accessed. The UED also provides a means for describing user characteristics and preferences such as the user's current geographic location and for describing the natural environment, such as illumination of the user's room. By facilitating descriptions of the user's environment, where the multimedia content is likely to be consumed, the UED contributes to maximizing the user's overall experience.

Content Provider: The properties of multimedia content vary widely and can hardly satisfy all the requirements of the broad spectrum of users. This creates a problem for the seamless and interoperable use of these resources and inorder to mitigate this problem, the framework provides a format for annotation of multimedia resources referred to as digital items. The MPEG–21 Digital Item Declaration (DID) provides a generic container format for associating metadata with multimedia content. This defines a requirement that the multimedia content will have appropriate resource description, using the Digital Item Declaration Language (DIDL), that provides details of properties, security and scalability points. Moreover, an independent concept from the provider is also defined, this is the Digital Item Adaptation (DIA) [46], it is the third component of the three–tier architecture and provides for the adaptation of the

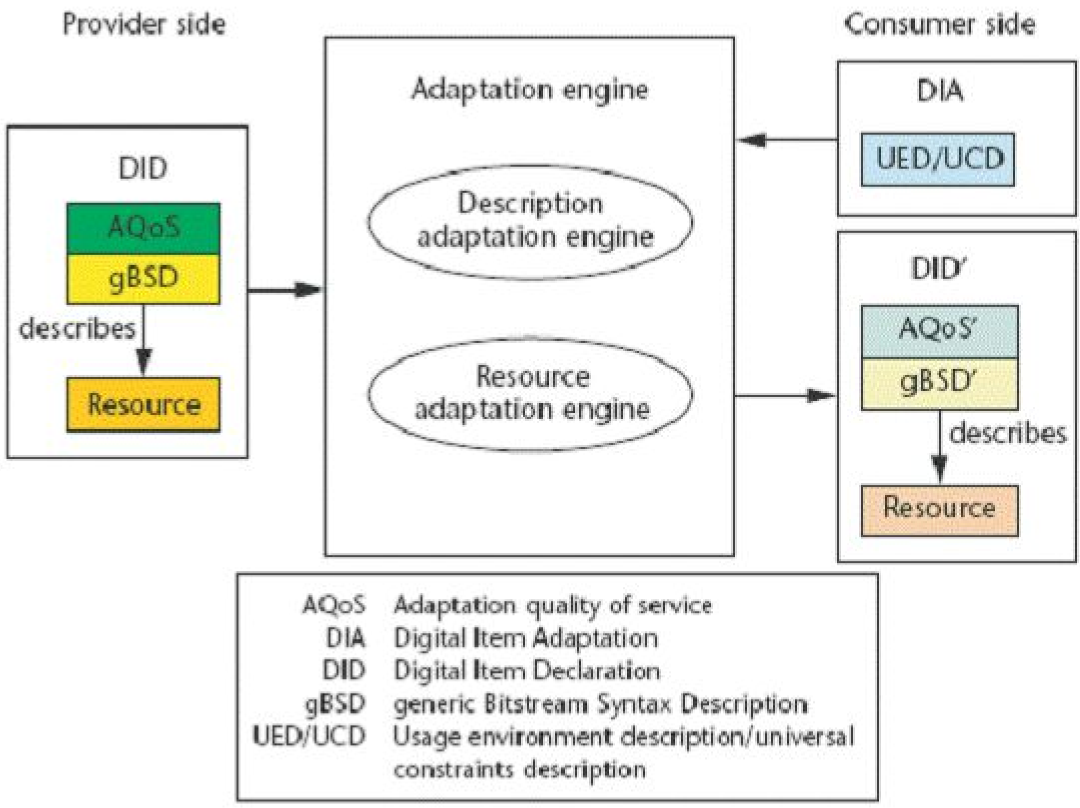

Figure 2.13: MPEG–21 Architecture [3]

Digital Items to the properties of usage environment.

However, the adaptation engines and their innovation is left open to researchers and developers while providing the standardized infrastructure. However, the adaptation is constrained to a set of tools for transforming the associated meta–data to reference different sections and properties of the resource that conforms to the requirement of the target usage environment. This is to enable a coding–format independent multimedia adaptation. The DIDs containing references to the actual multimedia resources, and their associated metadata, supply the provider–side input to an adaptation engine.

2.1.6 H.264/AVC

H.264/AVC (Advanced Video Coding) [47] is a digital video coding specification jointly developed by Moving Picture Experts Group of ISO/IEC and Video Coding Experts Group of ITU–T, in the so–called Joint Video Team (JVT), an alliance that unites the world's video standards coding experts in a single group. The work is based on earlier

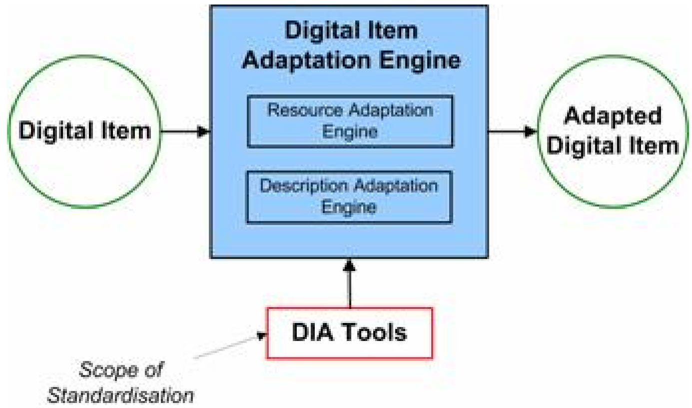

Figure 2.14: MPEG–21 Digital Item Adaptation [46]

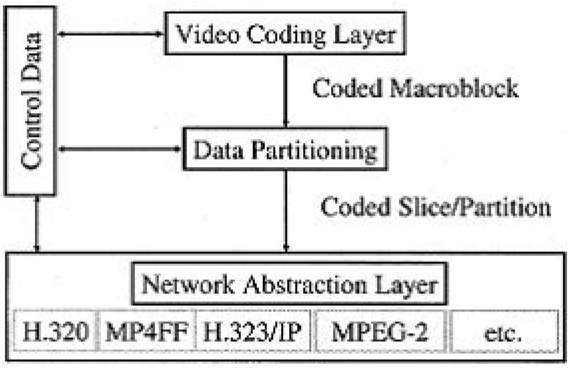

Figure 2.15: The Structure of H.264/AVC video encoder [47]

work in ITU–T codenamed H.26L. The specification for H.264/AVC were jointly developed and published as 'Recommendation H.264' and 'MPEG–4 part 10' by ITU–T and ISO/IEC JTC. The intent of H.264/AVC project was to create a standard that would be capable of providing good video quality at bit rates that are substantially lower than that of previous standards (MPEG–2, H.263 or MPEG–4 Part 2). And this should be achieved without so much increase in complexity as to make the design impractical (excessively expensive) to implement. An additional goal was to do this in a flexible way that would allow the standard to be applied to a very wide variety of applications (e.g. for both low and high bit rates, and low and high resolution video) and to work well on a very wide variety of networks and systems (e.g. for broadcast, DVD storage, RTP/IP packet networks and ITU–T multimedia telephony systems). The design of this video compression standard addresses the key functionalities of flexiblity, scalability and customizability which is required in applications of disparate use(s) and wide operating environments. It defines two operating layers: Video Coding (VCL) and Network Abstraction Layers (NAL). The Video Coding Layer (VCL) specifies the coding algorithms to efficiently represent the video content and the Network Abstraction Layer (NAL), formats the encoded video content for transmission across disparate networks or storage in a file system.

The outstanding features for multimedia communication includes but are not limited to the following:

1. Parameter set structure : This provides for robust transmission of picture header information that guarantee delivery under severe network conditions. The pictures would not be decodeable if the header information is lost.
2. NAL Unit Syntax Structure : The structure of the encoded video packet is customizable and thereby provides support for any underlying transmission network
3. Redundant Pictures : To provide support for robust video transmission, redundant pictures representing selected areas of the encoded pictures can be sent by the encoder.

2.2 Multimedia Adaptation Techniques

In pervasive media environments, users may access and interact with multimedia content on different types of terminals and networks. Such an environment includes a rich variety of multimedia terminals such as PC, TV, PDA, or cellular phones. One critical need in such an environment is the ability to handle the huge variation of resource constraints such as bandwidth, display capability, CPU speed, power, etc. The problem is further compounded by the diversity of user tasks ranging from active information seeking, interactive communication, to passive consumption of media content. Different tasks influence different user preferences in presentation styles and formats.

Video adaptation is an emerging field [48] that offers a rich body of techniques for answering challenging questions in pervasive media applications. It transforms the input video(s) to an output in video or augmented multimedia form by utilizing manipulations at multiple levels (signal, structural, or semantic) in order to meet diverse resource constraints and user preferences while optimizing the overall utility of the video.

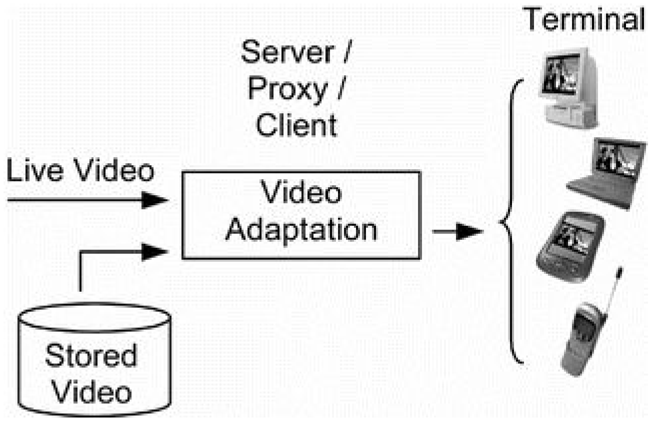

Figure 2.16: Video Adaptation to support heterogeneous terminals [48]

2.2.1 Scalable Video Coding

Scalable Video Coding [49, 50] provides a compression functionality whereby a bit stream is organized with a hierarchical syntax that enables a user to easily extract only a subpart of the data contained in the bit stream and still being able to decode the original input video but at a reduced spatial resolution, signal–to–noise ratio (SNR) or frame rate. This process can be applied recursively: once a new bit stream is extracted out of the original, it can undergo successive extractions corresponding to always lower properties of the scaled factors.

Scalable Video Coding addresses coding schemes for reliable delivery of video to diverse clients over heterogeneous networks using available system resources, particularly in scenarios where the downstream client capabilities, system resources and network conditions are not known in advance. For example, clients may have different display resolutions, systems may have different caching or intermediate storage resources and networks may have varying bandwidths, loss rates and best–effort or QoS capabilities.

There are several factors for scalability and each can be available with different granularity. The common factors are the temporal, spatial, quality and random access scalability. For example, if the original video signal contained 25 fps, temporal scalability would enable a client to decode a subpart of the bit stream generating a sequence with 10 fps or 5 fps. Spatial scalability corresponds to the possibility to access smaller spatial resolutions. For example, by decoding images with a resolution of 360x288 out of a video signal originally encoded at 720x576. Quality scalability represents the possibility to decode the bit stream at lower quality. In this case, the spatial and temporal resolutions remain the same, but the reconstructed signal will appear having more artifacts or less details. Finally, random access scalability corresponds to the possibility to access a region of interest out of the entire scene.

Several implementations of this technique exist, with the coding of a base layer and subsequent enhancement layers. MPEG defines [50] the following requirements for scalable coding :

1. Spatial scalability
This specifies the mechanism to support different levels of resolution, For example, resolutions from 44x38 to 3610 x 1536 in a single bitstream. In addition different levels of horizontal and vertical dimension ratios from one spatial layer to its spatial enhancement layer; addition of spatial information from one spatial layer to its enhancement spatial layer (e.g. 4:3 window in a 16:9 enhancement layer) is prescribed. For example, spatial scalability with resolutions 720x480 (SD) and 1280x720 (HD) should be supported, SD content being part of HD content.

2. Temporal scalability
This defines a required support mechanism that enables different levels of temporal scalability. Decoding of moving pictures with frame rates up to 60 fps should be supported. For instance, for Multi–channel content production and distribution, the same stream will be viewed on a variety of devices having different temporal resolution capabilities, for example, 7.5fps, 15fps, 30fps and 60fps should be supported.

3. Quality (SNR) scalability
This defines a required support for a mechanism that enables quality (Signal to Noise – SNR) scalability with finite (coarse–grained, e.g. 25% bit rate) steps. The decoded quality should vary between acceptable and visually lossless. Quality scalable coding should support a mechanism enabling medium–grained quality scalability (e.g. 10% bit rate steps). Quality scalable coding may support fine grain scalability (e.g. on a byte or MB level) and lossless coding.
Example :
In order to charge a different fee for higher resolution content (requiring more bandwidth/storage), different quality levels should be provided. An application for this scalability could be in the context of storage and transmission.

4. Complexity scalability
Complexity scalable coding specifies that the bitstream should be adjustable to the complexity levels and power characteristics of the receiving devices. Moreover, scalable video coding should enable the complexity to vary dynamically according to changing device characteristics.
Example :
A device can decide to trade–off the quality of the received video for a longer battery

life.

5. Region of interest scalability

Region of Interest (ROI) scalable coding should support a mechanism that permits interactive rectangular region of interest scalability with an access granularity of 32 pixels.

Example :

A user may desire that a certain region of the video should be displayed at a requested quality. Functions such as zoom are to be directed by the user (at the receiver end). Region of Interest scalability allows the user to both designate and obtain a region on the screen under the available bandwidth.

6. Combined scalability

Combined scalable coding defines a required support mechanism that enables arbitrarily combined quality (SNR), temporal, spatial and complexity scalability at least for a finite number of fixed points (these points representing any combination of spatial, temporal or SNR resolution, depending on the application). The appropriate number of fixed points will vary substantially according to the application (e.g. 8 for one application, 24 for another).

Example :

When a device moves from a high bandwidth to a low bandwidth connection, the change in the user experience should be gradual.

7. Robustness to different types of transmission errors

Scalable coding should be robust to different types of transmission errors.

Example :

For transmission over error–prone networks (wireless, Internet), the scalable coding should provide acceptable video quality under different types of error patterns (burst, independent, uniformly distributed, etc.).

8. Graceful degradation

Scalable coding should provide graceful degradation under different transmission error conditions.

Example :

For transmission over error–prone networks (wireless, Internet), the scalable coding should be able to provide acceptable quality video with graceful degradation.

9. Robustness under "best–effort" networks

Scalable coding should support a mechanism that enables robustness under "best–effort" networks, where all packets are treated equally and may be lost with equal probability.

Example :

For streaming over the conventional best–effort Internet or video over 802.11b/g best–effort wireless networks, the video can be coded using multiple description scalable coding to provide robustness under "best–effort" networks.

10. Colour depth

This defines support for colour depth scalability (e.g. extracting 8 bit content from a 12 bit coded bit stream or extracting the Y component (luminance) from a YUV coded bit stream) and the coding of moving pictures containing up to 10 bits per pixel component (linear and logarithmic). The standard may support coding of moving pictures containing up to 12 bits per pixel component (linear and logarithmic). Support for the coding of pictures in several colour formats or subsampling techniques is also required.

11. Coding efficiency performance

The embedded bitstream provided by scalable coding should not incur a larger coding efficiency penalty than 10% in bit–rate for the same perceived quality as compared with the bitstream provided by a single layer, state–of–the–art non–scalable coding schemes under error–free conditions. However this requirement might be relaxed in combination with other requirements (e.g. low–delay coding or combined scalability).

Example :

Unless the coding efficiency penalty for scalable coding is limited, scalable coding for transmission over bandwidth–limited networks will not be useful.

12. Base–layer compatibility

This defines a requirement to produce a base layer compliant with H.264/AVC. The issue of profiles/levels compatibility with H.264/AVC is to be considered, and may include any existing or forthcoming H.264/AVC profiles/levels.

13. Low Complexity Codecs

Scalable coding should enable low complexity implementations for encoding as well as decoding.

Example :

For service providers who need to create 'live' content, the complexity of the needed infrastructure needs to be minimized. For certain types of devices such as mobile devices it needs to be possible to implement encoders and decoders that work in resource–constrained environments. However complexity is always relative to implementation platform and decoded spatial resolution.

14. End–to–End Delay

This defines a requirement that scalable coding should support a low delay mode with a delay of no more than 150ms. In computing the delay, the frame period, encoding and decoding delay are to be included, but network/transport–related delay is to be excluded. When operating in low delay mode and compared to a non–scalable coder

configured for low–delay operation, the scalable coder shall not incur a larger coding efficiency penalty than 25% in bit–rate.
Example :
For conversational services or for changing channels when transmitting video over wireless networks, the end–to–end delay should be kept limited. This can be achieved with elimination of B–frames.
15. Random access capability
This defines a requirement that scalable coding should provide random access at any scalability layer (e.g. spatial, temporal, quality). It should support temporal random access with a granularity of less than 450ms.
Example :
When a user is switching between TV channels, random access should be provided. This can be achieved with proper insertion of I–frames.
16. Support for coding of interlaced material
For certain resolutions (SD – Standard Definition, HD – High Definition) scalable coding should provide a mechanism for coding interlaced material. The final bitstream may permit scalability between interlaced and progressive formats (e.g. ITU–R 601 interlaced material and CIF progressive material).
Example :
The coding of "broadcast" (ITU–R 601) would require coding of interlaced material.
17. System interface to support quality selection
A new scalable coding technology should define a uniform way to manipulate and adapt scalable streams that can be mapped easily to widely used protocols today. "Quality selection" specifies that the sending systems can dynamically generate different target frame rate, bitrate, and resolution from the stored bitstream based on usage environment.
Example :
The industry requires a standard interface to manipulate the scalable video contents created by the proposed technology via popular system layers and control protocols such as MPEG–4 systems, NAL units, MPEG–21 DIA, RTSP/SDP and SIP.
18. Multiple Adaptations
The scalable video bitstream should permit multiple successive extractions of lower quality bitstreams from the initial bitstream. It may permit non–linear extraction following different spatio–temporal paths.
Example :
When mobile video is being streamed over a 3G IP network with Multimedia Broadcast/Multicast Services (MBMS), the video bitstream can potentially be 'transcoded'

at various points in the network.

2.2.2 Transcoding

Format Transcoding : A basic adaptation process is to transcode video from one format to another, in order to make the video compatible with the new usage environment. This is not surprising when there are still many different formats prevailing in different application sectors such as broadcasting, consumer electronics, and Internet streaming.

One straightforward implementation is to concatenate the decoder of one format with the encoder of the new format. However, such implementations may not be feasible sometimes due to the potential excessive computational complexity or quality degradation. Therefore, reducing the complexity of the straightforward decoder–encoder implementation is a major driving force behind many research activities on transcoding.

What makes transcoding different from video encoding [51] is that the transcoding has access to many coding parameters and statistics that can be easily obtained from the input of compressed video stream. They may be used not only to simplify the computation, but also to improve the video quality. The transcoding can be considered as a special two–pass encoding: the "first–pass" encoding produces the input compressed video stream, and the "second–pass" encoding in the transcoder can use the information obtained from the firstpass to do a better encoding. Therefore, it is possible for the transcoder to achieve better video quality than the straightforward implementation, where the encoding is single pass. The challenge however is how to intelligently utilize the coding statistics and parameters extracted from the input to achieve the best possible video quality and the lowest possible computational complexity.

Spatial and Temporal Transcoding : The heterogeneity of communication networks and network access terminals often demand the conversion of compressed video not only in the bit rates, but also in the spatial/temporal resolutions. One of the challenging tasks in spatial/temporal transcoding is how to efficiently reestimate (or map) the target motion vectors from the input motion vectors.

Selection/Reduction : In resource–constrained situations, a popular adaptation approach is to trade some components of the entity for saving of some resources. Such schemes usually are implemented by selection and reduction of some elements in a video entity like shots and frames in a video clip, pixels in an image frame, bit planes in pixels, frequency components in transformed representation, etc. Some of these schemes are typically also considered as some forms of transcoding: changing the bit rate, frame rate, format or resolution of an existing coded video stream. Reduction involves a selection step to determine which specific components should be deleted. Uniform decimation sometimes is sufficient, while sophisticated methods further explore the non–equal importance of different components based on psychophysical or high–level semantic models. For example, in several video summarization systems, key events (such as scoring in sports) are defined based on user preferences or domain knowledge. During adaptation, such highlighted events are used to produce condensed video skims.

Replacement : This class of adaptation replaces selected elements in a video entity with less expensive counterparts, while aiming at preserving the overall perceived utility. For instance, a video sequence may be replaced with still frames (e.g., key frames or representative visuals) and associated narratives to produce a slide show presentation. The overall bandwidth requirement can thus be dramatically reduced. If bandwidth reduction is not a major concern, such adaptation methods can be used to provide efficient browsing aids in which still visuals can be used as visual summaries as well as efficient indexes to important points in the original video. Note that the replacement content does not have to be extracted from the original video. Representative visuals that can capture the salient information in the video (e.g., landmark photos of a scene) can be used.

Synthesis : Synthesis adaptation goes beyond the aforementioned classes by synthesizing new content presentations based on analysis results. The goal is to provide a more comprehensive experience or a more efficient tool for navigation. For example, visual mosaics (or panoramas) can be produced by motion analysis and scene construction. The extended view provides an enhanced experience in comprehending the spatio–temporal relations of objects in a scene. In addition, transmission of the synthesized stream usually requires much less bandwidth than the original video sequence since redundant information in the background does not have to be transmitted. Another example of adaptation by synthesis is the hierarchical summary of video. Key frames corresponding to highlight segments in a video sequence are organized in a hierarchical structure to facilitate efficient browsing. The structures in the hierarchy can be based on temporal decomposition or semantic classification.

In practical applications of adaptation, various combinations of the above classes can be used. Selected elements of content may be replaced with counterparts of different modalities, encoded with reduced resolutions, synthesized according to practical application requirements, and finally transcoded to a different format.

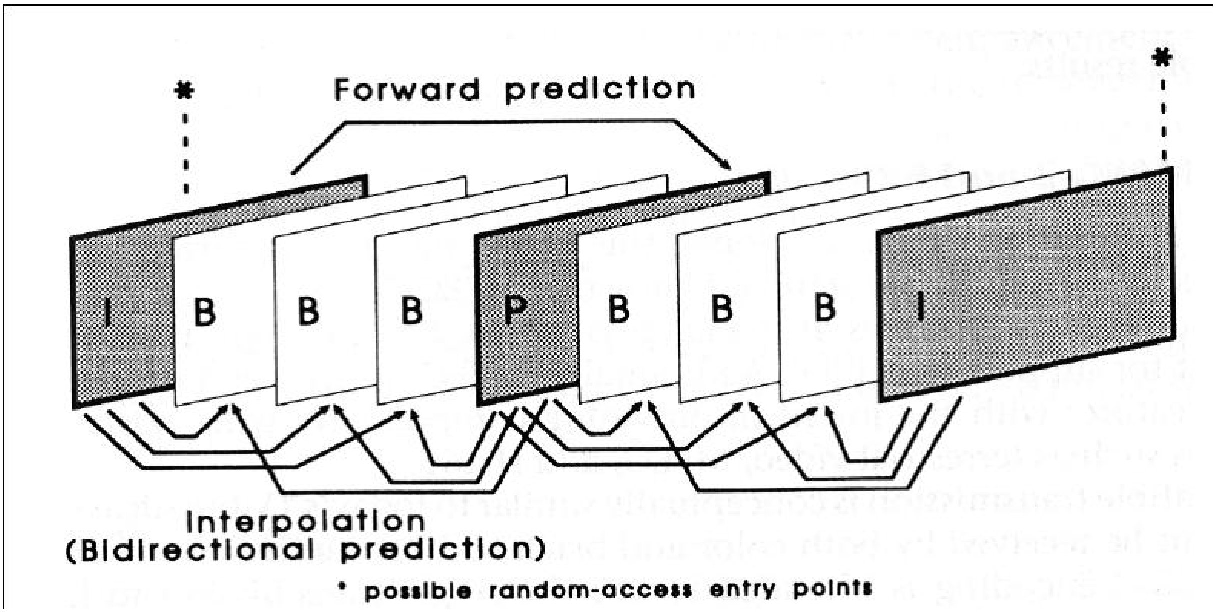

Figure 2.17: MPEG Frame types and display order [28]

2.2.3 Frame Dropping

Frame dropping is the most widely available and fastest way in terms of computational requirements to use temporal adaptation, which means to drop some frames. However only B– frames can be dropped. By dropping a frame at the server side and thereby not sending it, the needed bandwidth is decreased by the frames' size. Further, the destination terminal does not have to decode the unsent frames either.

As a favorable side effect, video compression systems using B–frames can produce smaller streams by leveraging motion detection and compensation over still images and additionally offer some amount of extra adaptability [52]. To offer good visual results at the client, it is advisable to drop frames uniformly distributed over time instead of dropping a number of consecutive frames. If the necessary means are available, visual results can be improved further with a priori information about the importance of certain frames. By this, only the visually less important ones are dropped. However the B–frames are the smallest in size in comparison to other frame types. If available network bandwidth stays low or decreases even more, frame dropping will reach its' limit. Only coarse–grained adaptation will help to overcome severe network problems. Frame dropping is only penalized with a more jerky visual experience. Increased visual experience can be provided [52] by introducing special temporal filters at the client side decoder, which interpolates intermediate frames.

2.2.4 Layered Multicast

Layered Multicasting is an application of Scalable Video Coding technique to manage user heterogeneity. In layered multicast, a sender of a session encodes an original stream into a base layer and several enhancement layers [53]. A base layer has the highest priority and higher enhancement layer has lower priority. The mechanism of adapting the number of received layers is referred to as rate control. In general, rate control for layered multicast is classified into two classes; receiver–driven scheme and network–supported scheme. In receiver–driven rate control, a sender transmits each layer on a separate IP multicast group. Each receiver joins some groups from the base layer and makes decisions to add or drop an enhancement layer. On the other hand, in network supported rate control, a sender transmits all the layers of a session on a single IP multicast group. Then routers adopt priority dropping mechanism which drops lower priority packets belonging to a higher layer, and decide the number of received layers for each receiver. Rate control for layered multicast must be designed to meet the following two requirements. First, rate control must prevent packet losses on a lower layer because they give more significant degradation of video quality. Second, rate control must assure fair bandwidth allocation among competitive sessions at a congested link.

Receiver–driven rate control has some limitations and cannot satisfy essentially the above requirements. In receiver–driven schemes, the network retains the uniform dropping mechanism such as drop–tail or Random Early Detection (RED), where all data packets are treated equally with respect to drop precedence. Although packet losses on a lower layer caused by random dropping lead to more significant degradation of received quality, receiver–driven schemes cannot but depend solely on packet losses to detect congestion because network has no mechanism to convey to receivers the explicit knowledge about the network conditions. Inevitably this causes packet losses on a lower layer. Receiver–driven schemes have another problem that receivers cannot respond to the congestion quickly because of both the delay for estimating network conditions and detecting congestion (estimation delay) and the delay for leaving a multicast group well known as IGMP leave delay. This slow reaction to the congestion cause more packet losses on a lower layer. In addition, many research [54, 55] show inability for receiver–driven schemes to achieve fairness among competing layered multicast sessions.

However traditional network–supported rate control, which adjusts transmission rate of each layered multicast session by priority dropping, also has two drawbacks. The first one is inability to achieve max–min fairness. According to the definition of max–min fairness [56], the excess bandwidth a session cannot use up because of the limitation on another link are allocated fairly among all other sessions. But, in a network supported scheme, a router cannot know the limitation on another downstream link. In order to assure fairness, signaling mechanism is necessary for routers to inform upstream ones in a multicast tree of available bandwidth at downstream links.

The second drawback is the transmission a fraction of a layer which is useless for receivers because generally in layered encoding scheme a layer is required to be received without any loss in order to be decoded. Transmitting this useless layer results in the waste of bandwidth on the path down to the receivers. Although priority dropping with the knowledge of the limitation can achieve max–min fairness, strict fair bandwidth allocation to each session causes receivers to receive the fraction of the highest enhancement layer among the layers received by receivers. In order to avoid transmitting the fraction a router needs a filtering mechanism which drops all the packets belonging to the highest enhancement layer, which has the lowest priority.

3 Mobile Multimedia Communication

Delivering multimedia services across mobile communication networks depends heavily on the properties of the access networks and overlying communication protocols. The access networks' properties like data throughput, error rates and quality of service underscore their capacity to support the delivery of these services. In addition, the properties and available services of the overlying protocols starting from the media access control to the application–level protocols cumulatively provides the defining limits of these services on the networks. These access networks and protocols that define mobile multimedia communication are presented in this chapter.

3.1 Mobile/Wireless Access Networks

Multimedia applications are increasingly being deployed to mobile communication networks, and this has in–turn impacted on the design and evolution of these networks. The performance of these networks depends on the efficiency and reliability of their access networks. The access network is defined[57] as a network entity providing access capabilities for various service applications and a wide variety of user terminals located at the edges of the networks.

Several innovations have been made at increasing the capacity of these networks to support any type of multimedia services – unicast, distributed and as well as interactive services.

Following the successes recorded in development of wireless technologies, the access networks has achieved data transfer speeds that supports real–time transport of multimedia data. However during the evolution[58] of these wireless infrastructures, a wide range of technologies have been developed, each with different capabilities and application areas, with no single technology good enough to replace all others. A review of the predominant wireless access technologies and their support for multimedia communication is presented in the following subsection.

3.1.1 GSM and 2.5G Networks

The Global System for Mobile Communication (GSM – originally Groupe Special Mobile) [59] is one of the well established and widely deployed mobile communication technology. However GSM was designed and optimized for speech communication, with limited support for data services with rates up to 9.6kbits/s full duplex. Support for data communication is provided through the traffic channels (TCH). Each connection uses one channel and the capacity of these channels is 13kbits/s for speech and 9.6kbits/s for data with possibility of an increase to 14.4kbits/s by changing the error protection coding. The GSM data–rates are too low for multimedia communication hence makes it effectively unsuitable for such uses.

However several enhancements has achieved higher rates for data services, these include the High Speed Circuit–Switched Data (HSCSD) and General Packet Radio Services (GPRS). HSCSD provides higher data rates by combining several traffic channels for a single connection with a maximum of seven (7) channels. In somewhat similar method, but with major enhancements to the GSM system; GPRS provides higher data–rates by combining maximum of eight channels. GPRS however requires some modification to the existing GSM network by introducing additional support elements. Despite providing improved data–rates, HSCSD and GPRS have not elicited expected patronage, this is because of the costly overhead of traffic channels required, which effectively reduces the available network bandwidth.

3.1.2 Universal Mobile Telecommunications System (UMTS)

Evolutions of the GSM mobile communication system, ranging from the previously mentioned HSCSD and GPRS to the different modes of EDGE (Enhanced Data rates for Global Evolution) culminated in an entirely new technology called UMTS (Universal Mobile Telecommunications System) which also maintains backward compatibility with the GSM. The UMTS also known as 3G, provides high data–rates of 384kbits/s in high mobility situations and up to 2Mbits/s in stationary user environment. The system uses the Wideband–Code Division Multiple Access technology for the (radio) access network. Several advances in this technology has recorded much higher access speeds. The High Speed Downlink Packet Access [1] (HSDPA) and High Speed Uplink Packet Access (HSUPA) technologies are evolutions of the W–CDMA technology with promising results in terms of high data–rates in both uplink and downlink. These technologies has the potential of offering data transmission speeds comparable to wireline networks like Ethernet. The UMTS provides a veritable platform for multimedia communication with robust support for global mobility and broadband connectivity.

3.1.3 802.11 – Wireless Local Area Networks (WLAN)

The IEEE 802.11 Wireless Local Area Networks [60] has gained prominence in recent times with wide acceptance and deployment. It has acquired a strong status in local area networking as an access technology of choice along side the once dominant wireline Ethernet (IEEE 802.3) technology. This is due to its' unique low deployment cost. WLAN technology follows Ethernet with the adoption of similar media access method (Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)) and similar data link protocol stack. The Direct Seqence (DS) or Frequency Hopping (FH) Spread Spectrum radio modulation was used on the unlicensed 2.4GHz frequency spectrum to achieve two specified raw data rates of 1 and 2 Mbits/s. However, several variations (IEEE 802.11a/b/d/e/f/g/h/i/x/p) resulting from amendments of the legacy standard exist and two widely deployed versions of the standard are 802.11 b and g.

IEEE 802.11b

The 802.11b amendment specifies new modulation technique called Complementary code keying (CCK) with raw data–rates of 11Mbits/s within indoor ranges of 30meters. However the capacity can be traded for range with a limit of 90 meters range at data–rate of 1Mbits/s.

IEEE 802.11g

In an effort to increase the data–rate, the 802.11g amendment specifies new modulation technique called Orthogonal frequency–division multiplexing (OFDM) with maximum raw data–rate of 54Mbits/s within indoor ranges of 30meters. However the range is marginally greater than that of 802.11b. In addition, the 802.11g maintains backward compatibility with 802.11a and b.

The WLAN access method requires that all user devices attempting a connection through a wireless access point (AP) would repeat access request on a random interrupt basis. This technique can result in distant user–devices being interrupted by closer ones, thereby reducing data throughput. Moreover the actual data–rates available to individual users are dependent on the number of users on an access point as it follows the principle of shared medium. This scenario has a defining impact on multimedia communication which depends on some guaranteed level of service, which will be difficult to maintain for large numbers of users.

IEEE 802.11e

In order to provide a guanranteed service, The 802.11e amendment provides enhancements for service differentiation, admission control, bandwidth reservation and link adaptation. The additional quality of service (QoS) mechanisms provides much needed support for delay–sensitive multimedia services.

3.1.4 802.16 – Wireless Metropolitan Area Networks (WMAN)

The IEEE 802.16 Wireless Metropolitan Area Networks [2] addresses connection in wireless metropolitan area networks, this technology with promising data–rates of 50Mbits/s over a range of 5Km, focuses on the efficient use of frequency bandwidth.

It defines a medium access control (MAC) layer that supports multiple physical layer specifications, customized for the frequency band of use. In the 802.16 MAC scheduling algorithm, connecting device(s) makes single data connection request, for initial entry into the network and thereafter is allocated a time slot by the base station. The time slot can be allocated or deallocated in accordance to their traffic conditions, but the connection is never re–assigned until the end of the session. This results to robust scheduling algorithm that maintains stability under severe load conditions. In addition the MAC has built–in mechanism for differentiated Quality of Service (QoS) to support the different needs of different applications. Since its approval in 2001, several amendments of the standard has been released, prominent among them is 'Revision 802.16e' otherwise known as 'Mobile WiMax' which specifies the use of scalable orthogonal frequency–division multiplexing (OFDMa) modulation technique to add enhancements which include coverage, self installation, power consumption, frequency re–use, bandwidth efficiency and a capability for full mobility support. The Wireless Metropolitan Area Network (WMAN) provides an inexpensive and easily deployable, strong platform for multimedia communication.

3.1.5 802.20 – Mobile Broadband Wireless Access (MBWA)

The Mobile Broadband Wireless Access Working Group [61] of the IEEE was established in 2002 to develop a technology standard (802.20) for fully mobile broadband wireless network. The development of this technology is still in progress as at the time of this thesis preparation and the scope of the standardization effort includes the specification of physical and medium access control layers, the design of an optimized datalink protocol for IP–data transport, with individual user data–rates per user in excess of 1 Mbits/s. It also specifies support for various mobility classes up to 250 Km/h in a MAN environment and frequency efficiencies that provides results comparatively higher than all existing mobile systems.

3.2 Transport Protocols

Multimedia transport systems provides the crucial functionalities of guaranteed data throughput and timely delivery, through specialized transport protocols. These transport protocols provide services that target different application environments, underlying media infrastructure and data link protocol types. For the internet, the TCP/IP protocol suite [62, 63] was designed and implemented to support simple data communication applications such as the file transfer protocol (ftp), electronic mail(e–mail) and terminal emulation(telnet). The two transport protocols TCP (Transmission Control Protocol) and UDP (User datagram Protocol) provided neither efficient nor real–time data transfer. In addition there was no multicast support. However as personal computers evolved and the Web (WWW) invented, the user domain enlarged with attendent need for online access to multimedia data. This trend resulted in the development of several protocols and support systems to supplement the legacy protocols in delivering the required functionalities for multimedia communication which includes reliability, flow control and Quality of Service. The section provides details of the major multimedia transport protocols.

3.2.1 MPEG Systems

MPEG–2 Transport Stream MPEG 2 Systems [64] (also presented in chapter 2) defines the transmission of multimedia streams comprising of one or more streams of video, audio and/or data, in a single or multiple stream. The most basic part of the MPEG bitstream is the Elementary Stream (ES) which contains either audio, video or data. The content of the elementary streams are organized into access units which would be a compressed frame for video. The elementary streams are grouped into Packetized Elementary Stream (PES) blocks with variable numbers of elementary streams but with a maximum size of 65536 Bytes. The PES contains a 6–Byte protocol header. This is broken into: 3–Byte start code, 1–Byte stream ID and a 2–Byte length field respectively. Each PES packet is broken into 188–bytes transport packets. The transport packets contains 184 Bytes payload and 4 Bytes header. The 32–bits (4 Bytes) Transport Packet header, starts with an 8–bits synchronization field, which is followed by a 3–bits flags, the bits of the flags indicates transport error, start of payload and priority packet respectively. After the flags is the 13–bits Packet Identifier (PID), this is used to identify the streams the packet belongs, providing the essential mapping between the TS, PES and ES. The next 2–bits are control bits used for scrambling and encryption, followed by two adaptation field control bits. The final 4–bits are continuity counter.

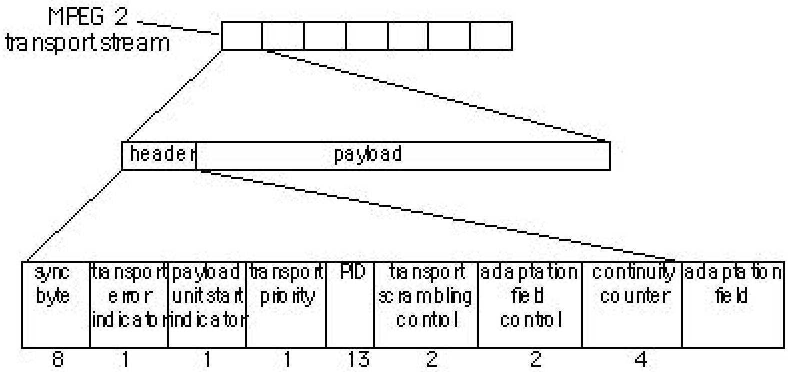

Figure 3.1: MPEG–2 Transport Stream Packet [37]

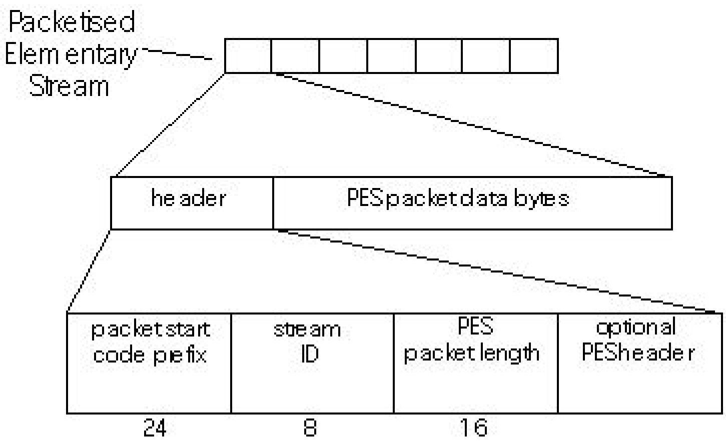

Figure 3.2: MPEG PES mapping onto the MPEG–2 Transport Stream Packet [37]

MPEG–2 Transport Stream uses short packet lengths (188 bytes) because it is intended for broadcast environments which are highly prone to error and the loss of one or more packets. These errors may be concealed or corrected by some decoding techniques. But the loss of a long packet, cannot be concealed easily. The Transport Stream can be used over a wide variety of media. It can be used over a lower layer network protocol, provided the protocol identifies the packet structure and updates the headers accordingly if errors occur. But the TS provides no reliability mechanisms hence is always used alongside other transport protocols, this results in duplicated functionality in computer networks as the Data Link layer features already exist.

MPEG–4 Delivery Multimedia Integration Framework

However, differently from its predecessor, MPEG–4 Delivery Protocol [65] was designed to adapt to multiple scenarios (local retrieval, remote interaction, broadcast or multicast) and delivery technologies. It abstracts the functionality of its delivery layer by creating a demarcation between the different pre–defined layers that manage only uniform co–related operations in a fashion that entrenches flexibility customizability.

The MPEG–4 layered model, comprises the Compression Layer, the Sync Layer and the Delivery Layer. The Compression Layer performs media encoding and decoding of Elementary Streams(ES – introduced in MPEG–2 TS); the Sync Layer manages synchronisation and hierarchical relations while the Delivery Layer ensures transparent access to content irrespective of delivery technology.

The delivery layer known as Delivery Multimedia Integration Framework (DMIF), has two important elements: the DMIF–Application Interface and the DMIF Signalling Protocol. The DMIF–Application Interface (DAI) is a semantic API that hides the details of delivery technology to the application and the DMIF Signalling Protocol is a generic session level protocol that is used for controlling multimedia data streaming. The protocol provides support for Quality of Service and resource management. the delivery protocol can be used on a wide variety of media and underlying protocol types.

3.2.2 RTSP

Real time streaming protocol (RTSP) [66] is an application–level protocol developed by the working group of the IETF responsible for Audio Video Transport for the control and delivery of multimedia data with real–time properties. The protocol supports a controlled, on–demand delivery of real–time data, such as audio and video. Sources of the multimedia data streams can include both live data feeds and stored files. The

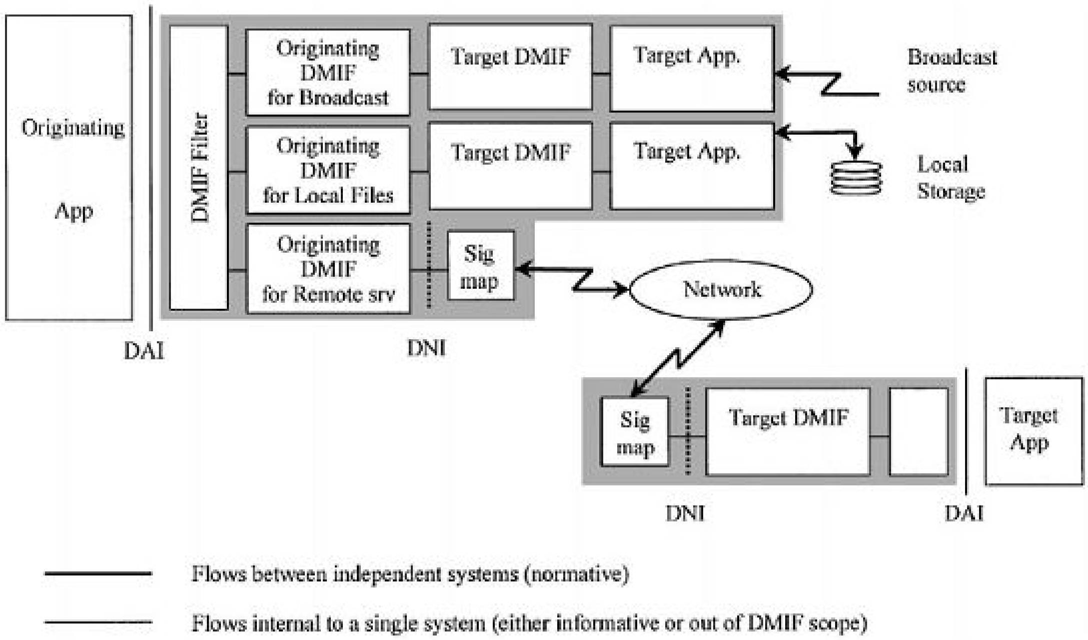

Figure 3.3: MPEG–4 DMIF communication architecture [65]

protocol is strictly for signalling and control of multimedia sessions and it provides a means for choosing delivery transport protocols such as UDP, multicast UDP and TCP. RTSP protocol specifies a two tier client–server communication architecture. In this setup, the server responds to multiple client connections and provides multimedia services. The client capabilities includes playing, recording, setting to fast forward, rewinding, and pauseing of multimedia data streams. It can also request specific delivery parameters which includes the IP addresses, UDP ports and codecs. RTSP is a text based protocol and specifies the message formats and their functionalities. The messages are divided into two classes 'requests' and 'responses'. The standard specifies several request methods which includes the following: DESCRIBE, ANNOUNCE, GET PARAMETER, OPTIONS, PAUSE, PLAY, RECORD, REDIRECT, SETUP, SET PARAMETER and TEARDOWN. The response messages has the following classifications

1xx: Informational - Request received, continuing process
2xx: Success - The action was successfully received, understood, and accepted
3xx: Redirection - Further action must be taken in order to complete the request
4xx: Client Error - The request contains bad syntax or cannot be fulfilled
5xx: Server Error - The server failed to fulfil an apparently valid request

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

Figure 3.4: RTP Packet Header Format

3.2.3 Real–time Transport Protocol (RTP)

The Real–time Transport Protocol (RTP) was developed by the working group of the Internet Engineering Task Force (IETF) responsible for Audio/Video Transport. RTP provides solution to the problem of real–time data transfer on TCP/IP protocol suite. TCP transport protocol does not support real–time communication because packets losses are handled with re-transmissions and acknowledgements in order to provide reliable and sequential data flow. This effectively introduces delays hence it cannot guarantee timely delivery of data.

However UDP transport protocol avoids this problem with a simple datagram data transfer method with no guarantee of ordered delivery or error recovery. This provides support for real–time communication but the service is very unrealiable. RTP is used to augment the UDP transport protocol by defining application–level data packet formats with time–stamps, sequence numbers, payload types, synchronization sources, contributing sources and other control information. The payload types define the formats of the multimedia data and also supports user defined payloads using the 'dynamic payload'. The Real–time Transport Protocol enables applications to provide reliable and real–time audio/video data transport using UDP. The packet format of the RTP protocol is presented in table 3.1:

Field	Size	Description
version (V)	2 bits	This field identifies the version of RTP. The current version defined is two (2).
padding (P)	1 bit	If the padding bit is set, the packet contains one or more additional padding octets at the end which are not part of the payload.
extension (X)	1 bit	If the extension bit is set, the fixed header is followed by exactly one header extension.
CSRC count (CC)	4 bits	The CSRC count contains the number of CSRC identifiers that follow the fixed header.
marker (M)	1 bit	The interpretation of the marker is defined by a profile. It is intended to allow significant events such as frame boundaries to be marked in the packet stream.
payload type (PT)	7 bits	This field identifies the format of the RTP payload and determines its interpretation by the application.
sequence number	16 bits	The sequence number increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence.
timestamp	32 bits	The timestamp reflects the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations
SSRC	32 bits	The SSRC field identifies the synchronization source.
CSRC list	32 bits	0 to 15 items, 32 bits each

Table 3.1: RTP Packet Header Format

3.2.4 Real–time Transport Control Protocol (RTCP)

The Real–time Transport Control Protocol (RTCP), was developed to be used along side the RTP to transfer control information which includes transmission data statistics, design specific information and service request by communicating parties in an RTP session. Several RTCP packet types are defined which includes Sender's Report (SR) – which provides statistics of the RTP session from the sender, Receiver's Report (RR) – provides statistics of the RTP session from the receiver, Session Description (SDES), Bye message (BYE) and Application specific message (APP).

3.3 Session Protocols

Multimedia communication sessions can be initiated in different modes which includes unicast, multicast, broadcast and a hybrid of the three(3) groupings. The protocol for the management and co–ordination of these sessions are presented in this section.

3.3.1 Session Description Protocol

Session Description Protocol (SDP) was developed by the working group of the Internet Engineering Task Force (IETF) responsible for Audio/Video Transport, to provide a common standard for description of multimedia session. The description includes name and purpose of the session, media types, timing and protocol information. it is intended for use in multimedia sessions with wide participation. However the description can be transported by any underlying transport protocol as it does not include one. The syntax of the description is presented:

SDP Syntax

An SDP description consists of lines of text in the format:

<type>=<value>

List of the syntax is presented in table 3.2:

SDP Key	**Description**
v=	Protocol Version
o=	The Owner/Creator and Session Identifier
s=	Session Name
i=*	Session Information
u=*	URI of description
e=*	Email Address
p=*	Phone Number
c=*	Connection Information
b=*	Bandwidth Information
z=*	Time Zone Adjustments
k=*	Encryption Key
a=*	zero or more Session Attributes
t=	Time the Session is Active
r=*	zero or more Repeat Times
m=	Media Name and Transport Address
i=*	Media Title
c=*	Connection Information
b=*	Bandwidth Information
k=*	Encryption Key
a=*	zero or more Media Attributes
	"*" indicates an optional field.

Table 3.2: SDP Syntax

Request Method	Purpose
INVITE	Invite the callee into a session
BYE	Terminate a call or call request
OPTIONS	Discover the capabilities of the receiver
ACK	Acknowledge a successful response
CANCEL	Terminate incomplete call requests
REGISTER	Register the current location of a user

Table 3.3: Six core request methods

3.3.2 Session Announcement Protocol

Session Announcement Protocol (SAP) was developed by the working group of the Internet Engineering Task Force (IETF) responsible for Audio/Video Transport, to provide a common standard for announcement of multimedia sessions. In this protocol, the session creator transmits SAP packets to a pre–defined multicast addresses, the SAP packets contains description of the session in SDP (Session Description Protocol) format. The multimedia clients listens to this multicast addresses and receives the periodic announcement of sessions. The protocol defines a packet header and an SDP payload, in addition security and caching are also taken into account.

3.3.3 Session Initiation Protocol

Session Initiation Protocol (SIP) was developed by the working group of the Internet Engineering Task Force (IETF) responsible for Audio/Video Transport, to provide a moderate and simple, common standard for initiation of multimedia sessions on the internet. The protocol specifies procedures for setup, management and teardown of telephone calls, video conferencing and other multimedia connections, using an ascii text based message format. The SIP messages are divided into 'REQUEST' and 'RESPONSE' messages. The request methods are used by the caller to establish, manage and/or teardown a connection while the responses are used by the callee to respond to the received request. The six core request methods and response codes are given in table 3.3 and 3.4.

Response Class	Purpose	Example
1XX	Information about call status	180 RINGING
2XX	Success	200 OK
3XX	Redirection to another server	301 MOVED TEMPORARILY
4XX	Client did something wrong	401 UNAUTHORISED
5XX	Server did something wrong	500 INTERNAL SERVER ERROR
6XX	Global failure	606 NOT ACCEPTABLE

Table 3.4: Six core response classes

3.3.4 H.323/H.324

The H.323 protocol suite was developed by the ITU–T to provide internet telephony. The protocol model provides specification for interconnection between the Internet and Public Switched Telephone Network (PSTN), through inter–operation of several other specific protocols for call setup, signalling, codec negotiation and data transport. The standard known as ITU–T Recommendation H.323, "Visual Telephone Systems and Equipment for Local Area Networks which Provide a Non–Guaranteed Quality of Service" provides for audio–visual communication (from a desktop computer to any terminal equipment), the standard is a specification of several other ITU–T recommendations which includes the following: H.225.0 which provides a description of the packet and synchronization formats of the multimedia and control streams. H.245 provides description of messages and procedures for initiating and terminating multimedia stream sessions, exchanging terminal equipment capabilities and other control operations between two communicating end–points. T.120 provides description of general requirements for data applications. H.323 specifies the required multimedia codecs for communication, for audio, G.711 and its later variants while for video, H.261 and all versions of H.263 and H.264. More so the use of Real–Time Transport Protocol (RTP) for encapsulating the audio–video payloads, and the Real–Time Control Protocol for transport of control statistics and other required data is also specified.

H.324 : ITU–T Recommendation H.324 [67] is a standard for multimedia conferencing on low bitrate circuit switched networks, it provides for interoperability of a variety multimedia terminals which includes PC–based multimedia videoconferencing systems, telephones, World Wide Web browsers with live video, remote security cameras, and standalone videophones. The H.324 standard was built on an earlier ITU–T standard

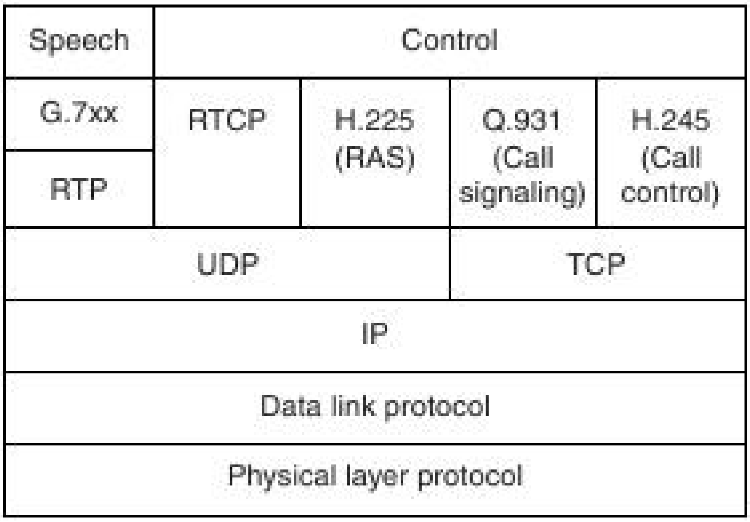

Figure 3.5: H.323 Protocol Stack [66]

for ISDN videoconferencing ITU–T H.320, and hence shares basic architecture, which consists of a multiplexer for mixing the various media types into a single bitstream (H.223), audio and video compression algorithms (G.723.1 and H.263), a control protocol for negotiation and logical channel control (H.245), H.233/234 encryption and H.224/281 far–end camera control. It also has requirements for V.34 modem as it was primarily designed for General Switched Telephone Network (GSTN) and specifies the use of mandatory V.8 or optional V.8bis protocol for call startup, to identify the modem type and operation mode.

4 System Design

The task of providing a seamless connection between internet multimedia services and mobile delivery platforms presents two outstanding challenges:

1.) Incompatibility of properties/formats of these multimedia services and that of the mobile devices. There exists temporal, spatial, format and encoding rate incompatibilities.
2.) Diversity in users' service levels and access technology types. The service levels of individual users vary widely ranging from "guaranteed service" to "best efforts". In addition, in heterogeneous environments, the access technology properties vary according to the available access networks.

In solving the first problem of incompatibilities, the best and easiest method would be to equip the device to perform the required conversions. The spatial resolution can be re–mapped to fit the devices' display without incurring so much computational cost. The temporal and rate problems would be much more difficult to solve as the task of decoding the media streams at very high rates may have severe processing costs. Moreover, format conversion problem would prove to be the most difficult as this requires mobile devices to have the installed capability of decoding every available multimedia format. The memory, power and processing cost of this requirement though achieveable is extremely high. The second problem of diversity in users' service level and access technology properties cannot be solved at the client level. The mobile device has no control over the network transmission path therefore relies on the services of these access networks for delivery of the multimedia streams.

In error–prone or best–efforts networks, the delivery of the multimedia services cannot be guaranteed, however buffering strategies and other techniques can be explored to augment the properties and available capacity of the access network. But this cannot in itself guarantee that the multimedia streams would be without errors if delivered at all. Therefore an external supporting mechanism is required to provide an effective solution. In line with this background, a single approach that attempts to optimally combine and solve this two problems is presented in the following sections.

4.1 Network Delivery Architecture

The multiplicity of mobile devices and diversity of their characteristics creates a need for an optimally functional infrastructure of network elements that supports real–time delivery of multimedia services. Currently it is not possible to provide a single stream that will correctly (let alone optimally) satisfy all the variable user, device and access technology dependent parameters. Therefore a need to adapt these streams at some point to the specific user needs and characteristics is created. Current research in this direction provides three options of architecture model for multimedia content adaptation: server side adaptation (the content provider adapts the stream before transmission), proxy adaptation (the adaptation occurs at a point, prior to delivery to the user terminal); and client adaptation (the adaptation occurs at the client terminal). The identified overall benefits and drawbacks of these methods are itemized in table 4.1.

The three(3) models have their benefits and drawbacks but the proxy server adaptation model provides support for dedicated services such as optimal implementation of complex transformations, processing/storing of user profiles/personal preferences, prediction of traffic patterns and forecast of resource requirements. In performing an efficient and cost–effective real–time adaptation of multimedia content, proactive techniques in addition to reactive solutions will enhance overall performance and such an elaborate structure will require the dedicated services provided by this model. In addition significant section of the research community validates this approach as presented in [68], and the MPEG–21 universal multimedia access is based on this structure. Therefore this model is adopted for the adaptation architecture.

In line with the adopted proxy model and within the context of mobile delivery terminals, the positioning of the proxy is of extreme importance to the overall achievement of the design objective. A typical scenario of mobile network operators providing service to its user community as depicted in figure 4.1. In this scenario, a proxy server within each operator's network would enable each user to freely and independently access the desired multimedia content from several content providers. This provides the much needed personalization (user/device) and context–awareness.

However, mobile devices are increasing multi–homed, offering users great choice and freedom in selecting and roaming across variety of wireless access technologies, thereby varying cost of service, application availability and quality of service. This introduces another level of complexity to the delivery architecture as maintaining the operator based proxy approach in this scenario (depicted in fig 4.2) would be excessively difficult to implement and completely impractical to operate due to the following factors :

MODEL	BENEFITS	DRAWBACKS
Client Side	– The Client has full information on own own capabilities, user preferences and usage context. – The client always has up-to-date information.	– Requires Terminal processing resources (CPU, battery life). – Wasteful traffic and sometimes not possible.
Server Side	– Full control to the content. – Pre calculated content representations possible.	– Infrastructure maintenance. – Providing all possible variations of streams greatly increases cost of delivered content.
Proxy	– Moves computation from the client and server site to the proxy. – Dedicated services possible.	– Possible bottleneck. – License and right issues over data.

Table 4.1: Adaptation Architecture Models

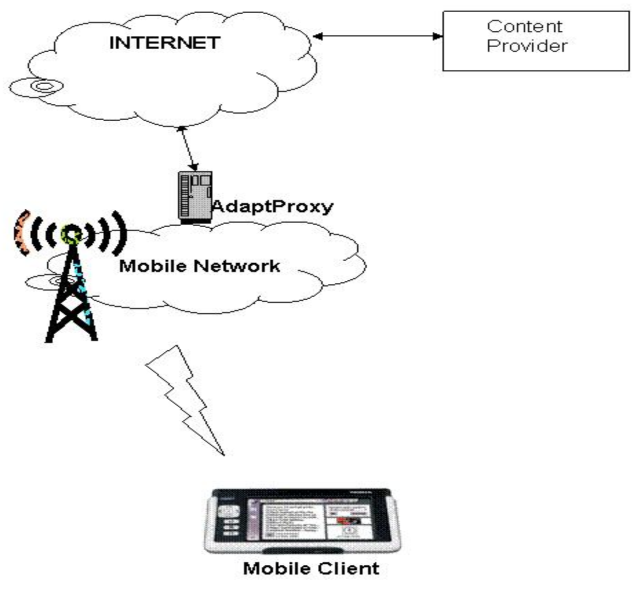

Figure 4.1: Mobile/Wireless operator based proxy approach : Accessing online multimedia content from mobile/wireless networks

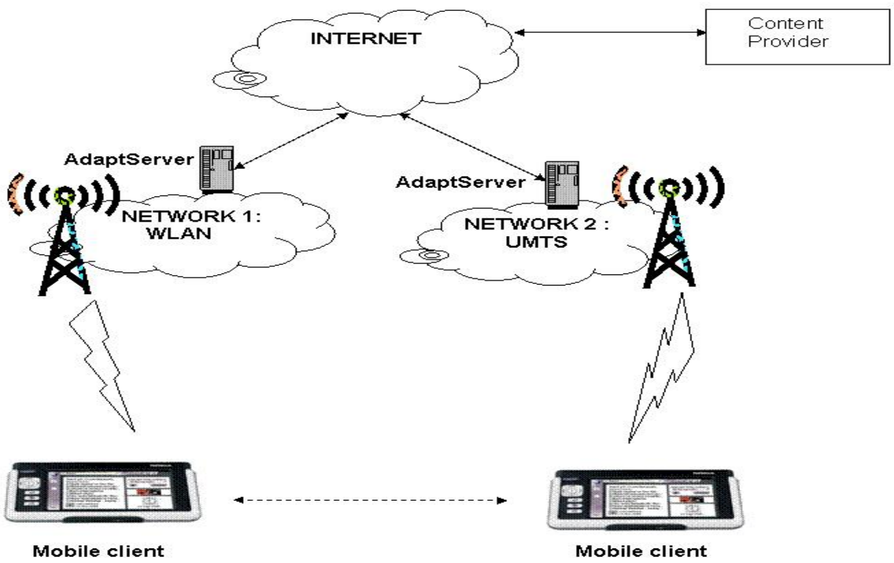

Figure 4.2: Mobile/Wireless operator based proxy approach : Accessing online multimedia content from mobile/wireless networks on multi–homed devices

1. Deployment of standard proxy servers across disparate mobile/wireless networks (both public and private, for example : WLAN) can neither be guaranteed nor enforced.
2. Distribution of user/device information on multiple proxies with no specific pattern of utility is wasteful.
3. Maintaining of continuous multimedia data streams across several proxies of different load conditions would be difficult to achieve.
4. Maintaining an acceptable overall quality of user experience would be difficult to achieve.

Following the pitfalls identified in the operator based proxy approach, a virtual proxy service is proposed. In this network delivery architecture, the proxy server is positioned as a virtual service, (depicted in fig. 4.3) the mobile user can connect to the same proxy through any mobile/wireless network. The proxy uses detailed device information which includes the available access technologies and their defined (desired, contractual or default) service types and levels to maintain context–awareness in the delivery of multimedia services.

4.1.1 Application Use – Case

The application framework is divided into two separate and independent components, the proxy server and the mobile client. However, a virtual proxy server creates a

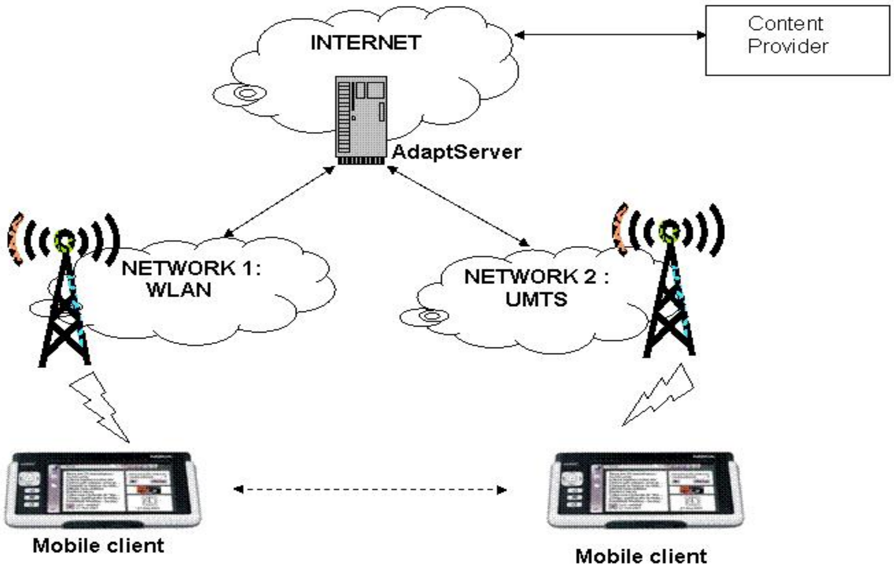

Figure 4.3: Virtual proxy approach : Accessing online multimedia content from mobile/wireless networks on multi–homed devices

problem of applicability of the solution as none of the identified components (content provider, mobile operator and content consumer) bears any responsibility to the provision of the proxy services. Therefore an application use case is proposed as follows: First, a potential user, downloads and installs the client software on a mobile device, the installed software sets up the user profile, the connection parameters and updates the proxy. Finally the user is able to access any multimedia streams located anywhere on the internet. The host of the proxy might provide multimedia services, adverts or any other forms of intermediary services.

4.1.2 Adaptive Delivery Framework

The Adaptation proxy performs multimedia content adaptation, this is the transformation of the input multimedia data to data conformant to target usage environment and prevailing network conditions.

$$\mathrm{D} = \mathrm{x}D_0$$

where x – represents the transformation factors : spatial, temporal, formats and quality.

This transformation process can be broken down into the following sequences of processes.

$$1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 4 \longrightarrow 5 \longrightarrow 6 \longrightarrow 7$$

The processes are :
1.) Source : Receiving the multimedia streams from source(s) specified by the mobile users. These could be live or pre–stored sources of multimedia data.
2.) Decoder : Decoding the input multimedia data to raw formats.
3.) Spatial Transformer : Performing scaling of the decoded streams to the supported resolutions by the mobile device.
4.) Temporal transformer : Performing temporal transformation of the spatially transformed streams to the supported framerate by the mobile device.
5.) Format transformer : Encoding the transformed data into the supported format and bitrate (same as quality).
6.) Transport format packetizer : Encapsulating the encoded multimedia streams in the transport format supported by the mobile device.
7.) Network Transmitter : Transmits the encapsulated media to the mobile device.

The parameters required at the transformation stages (3, 4, 5) is determined by the device profile. The proxy maintains individual profiles of all connecting mobile devices, through the client software on the devices. The profile contains detailed information of the devices' supported multimedia formats, available access technologies and their service types. The service type on an interface could be guaranteed service with a defined data transfer rate or best effort. The proxy streams a constant bitrate for guaranteed service or triggers the rate control mechanism that monitors the traffic and signals rate changes. At the packetization stage (6) the multimedia data can be encapsulated as RTP or MPEG–2 TS (and RTP if defined) packets and sent over the UDP/IP stack.

UDP is used because TCP service limits the possibility of real–time delivery. The proxy/client system provides an adaptive framework for delivery of multimedia services. The mobile client routes all service requests to the proxy server and receives multimedia services adapted to its' usage environment defined in the profile. This design closely follows the proposed MPEG–21 usage environment description profile.

The transformation process is absolutely necessary as the formats of the input multimedia cannot be determined priori. However the process incurs relatively high computationally cost, as the process requires the instantiation of this chain of events for each user request. For pre–stored sources, processing cost could be reduced by applying a processing scheduling scheme, where each adaptation instance is allocated a processing time slot. This scheme would require buffering strategy at the client in order to maintain a smooth playout. However, for live sources, the computational cost is inevitable. In addition the processing cycles of the transformation elements introduce delays which results in temporal shift of the overall multimedia streams. The processing, packetization and transmission delays should not exceed the prescribed real–time threshold. This effectively places strong computational requirements for adapting live streams and satisfying real–time constraints.

4.2 Network Rate Control

Streaming multimedia data across heterogeneous access technologies would require adaptive techniques to support the different data rate characteristics of these access networks. In–addition best efforts networks provide no guarantee of availability of data traffic bandwidth, therefore support for variable traffic capacities across interfaces is required.

In line with this premise, two strategies were adopted to provide adaptability, these include : source rate , network rate and congestion control

I – Source Rate Control
Multimedia data (video and audio) is essentially encoded at specific rates, this determines the volume of data produced at the encoder. Making a brief analysis with uncompressed video with resolution of 704 x 576.

For 25 frames per second frame–rate:
1 frame of video (24bits to store each pixel) at 25fps = 243Mbits/s.
Therefore video compression is used to reduce the quantity of data generated and the source rate is controlled by changing the compression parameters, two methods were used : 1.) GOP Structure and 2.) Quantization

1. GOP Structure: A parameter called Group of Pictures size (gop–size) is used to determine the pattern of frame type selection during encoding of the video.

A basic analysis of the impact of the technique :
For example: assigning a gop–size of 15 and assuming (quite credibly) that P–frames and B–frames would reduce to fifty percent(50%) and twenty five percent (25%) of original frame respectively. The encoding pattern would result to: IBBPBBPBBPBBPBB

pattern of fifteen frame cycle. For 25–frames per second, the resulting bitrate without any other form of compression will be 92Mbits/s

The source rate of the video encoding process can be changed by varying the gop–size parameter in a multiple of 3 (maximum of 15) and also to significantly reduce the source rate, some percentage of the B–frames could be dropped prior to transmission but this has strong limits and severe side effects so it was not used in this design.

However for lossy networks like wireless links, any loss of the I–frame or P–frame will result in severe errors as the B–frames cannot be decoded independently of the I– or P– frame. This will propagate errors all through the chain of B–frames received until another I– or P– frame is received. Moreover the display pattern is different from the encoding pattern. The video encoder pattern is : IPBBPBBP and that of the decoder is : IBBPBBP, this requires that the frames received are buffered before play out. This further places a memory constraint and introduces delay on mobile devices. Therefore this feature is optional (the proxy sets the gop–size to 1 by default) and can be used if reliable connection and the memory capacity on the client exists.

2. Quantization : Reduction of the quantity of data stored per pixel, it is used to constrain the sizes of DCT (Discrete Cosine Transform) co–efficients. Quantization can be performed using the different step size for a macroblock (i.e., using a non–uniform quantization matrix). However, for the first DCT coefficient of each intra block (DC coefficient for blocks from intra–frames) a different quantization step size is used. Most compression systems has a parameter 'bitrate', specified in bit per second. The compression algorithm achieves this target bitrate by calculating the appropriate quantization step size per block. Increasing the quantization step size, increases compression thereby reducing the bitrate and quality as well. To acheive a target source rate, the following formula was used:

For video :

$$Sr = (W * H * Fps * Cp * Gp * Qf)$$

where W – is the width of the video
H – is the height of the video

Fps – the framerate
Cp – is the Compression parameter for I–frame (measured per codec at the best quality, ratio of raw frame size to compressed frame size)
Gp – is the Compression factor for each Gop–size (measured per codec by segmenting a given video sequence using shot detection and encoding the inter–frames, the value is calculated from the Cp of each frame type, This is the ratio of the sum of the Cp per frame type to the Gop Size)
Qf – is the quantization factor (0 < Qf <= 1)

For audio :

$$Sr = (N * SS * SF * Cp * Qf)$$

where N – is the number of channels
SS – is the sample size
SF – Sample Frequency
Cp – is the Compression parameter (measured per codec at the best quality)
Qf – is the quantization factor (0 < Qf <= 1)

II – Transmit Rate Control

Source rate defines the rate at which the video encoding application generates data, this is also the expected rate for playout at the mobile terminal. However, the network transmit rate is the data rate over the transport layer that supports the data throughput specified by the source rate at the receiving terminal. To transmit the multimedia data several overhead information in added to the fragments of the data, therefore frames transmitted at the physical layer contains significant level of overhead that limit the effective throughput of the medium. In order to guarantee the delivery of multimedia data at the source rate, a data framing strategy believed to efficient/effective is proposed. The scheme is a method of calculating the application and transport protocol frame size(s) and the adjustment of these sizes to sustain the transmission of the encoding rate and support the characteristics of the network layers. Analysis of the scheme is presented:

RTP application layer framing is used to encapsulate the multimedia data before sending them onto the lower layers of UDP and IP. Beneath UDP/IP is a possibility of access network types between the proxy and the mobile device.

However on the application and transport level:

To calculate the Transmit Rate Tr for a given source rate Sr

$$Tr = (IPd + IPh + RTIPh)\,\frac{Sr}{IPd}$$

where
IPd – is the actual data size in the IP payload
IPh – is the IP header size (UDP/IP)
RTIPh – the RTP header size per IP packet

and RTIPh = (RTh x IPd)/RTd

RTd – is the RTP payload
RTh – is the RTP header size

$$Tr = (IPd + IPh + \frac{(RTh * IPd)}{RTd})\frac{Sr}{IPd}$$

For a source rate of 1.2Mbits/s and RTP packet size of 1024bytes

IPd = 1024bytes, IPh = 28bytes (UDP/IP header), RTh = 12bytes

Transmit Rate Tr = 1.246Mbit/s

A graph of the Transmit Rates Tr and packet sizes at three different source rates is presented in the appendix A.1, A.2 and A.3.

III – Jitter and Congestion Control

On the client the packets arrive out of order (characteristic of UDP Transport) and the inter–arrival times are irregular. This scenario known as Packet Jitter requires a play out strategy on the mobile client. For an ideal situation, the client plays out the multimedia data streams as it arrives thereby maintaining an ordered and steady flow of multimedia data streams. To maintain this steady flow, the received multimedia data is buffered and re–ordered prior to playout. However, in some cases the jitter level varies widely, this could result in inadequate buffer sizes for handling data re–ordering. To eliminate this problem, a buffering strategy is adopted where the size of the buffer is dynamically adjusted according to the calculated inter–arrival time variance.

$$Inter - arrivaltimevariance = \frac{\sum_{(t-tm)^2}}{N}$$

where
t = the difference in arrival time of a packet
tm = the mean of the differences in a sample
N = total number of packets in a sample

This requires the continuous sampling of packets within a specified time threshold and re–calculation of the variance to adjust the buffer size. However, client buffering, increases the resource requirement in terms of processing and memory demands. In addition, the buffering introduces delays to the multimedia streams.

Congestion of network due to competing user access to shared media is inevitable in best effort networks and the only solution to contain losses due to congestion is to throttle the source encoder and reduce the encoding rate. The first step in achieving this objective is to detect delay on the end to end link and hence determine the transmit rate (Tr).

To calculate the delay:

A probe packet is sent from the proxy to the mobile client. T1 is the time the packet was sent, and the client records the time it received the packet as T2 and returns the packet to the proxy alongside with T2 and T3 (the time the return packet was sent). The Proxy receives the packet and records the time it received it as T4.

On the forward path

$$T2 = T1 + d1 + offset$$

where d1 – is the delay on the forward path and offset is the difference on the two clocks.

On the reverse path

$$T4 = T3 + d2 - offset$$

where d2 – is the delay on the reverse path and offset is the difference on the two clocks.

Let D = The average of the delays

$$D = \frac{(d1 + d2)}{2}$$
$$D = \frac{(T4 - T3 + T2 - T1)}{2}$$

However this method cannot be used in networks with different uplink and downlink speeds. Another method that satisfies this criteria is presented.

To calculate the delay in one direction:

Five probe packets are sent from the proxy to the mobile client. The sizes of the probe packets are multiples of the media frames sizes i.e. Sizes of Packets 1, 2, ..., 5 = 1, 2, ..., 5 x (maximum media frame size)

Time T is the time the packet was sent and the client records the time it received the packet as TT.

Hence for the five packets, T1, T2, T3, T4 and T5 are the times the packets were sent while TT1, TT2, TT3, TT4 and TT5 are the times the packets were received.

The time difference D = TT - T is the delay on the forward path and offset of the two clocks.

For the five packets : D1, D2, ..., D5 will form an arithmetic progression due to the resultant delays from the increasing packet sizes.

This gives

D1 = a + d, D2 = a + 2d, ..., D5 = a + 5d

where a = clock offset and d = delay.

Using the first two values D1 and D2 the offset and delays can be calculated. Thereafter, this is used to generate the next values. The generated values are compared to the measured values with the mean deviation used to adjust the delay value. For greater sensitivity, the number of probe packets can be increased to 10, 15, ..., 30. The method can be used in either direction, but it was not implemented.

4.3 Multi–Homed Delivery

The availability of multiple wireless network interfaces on mobile devices, makes a case for the development of novel techniques that will enable the simultaneous use of these interfaces. This solution will provide several benefits which includes:
1. Robustness : The use of multiple interfaces will effectively reduce (if not entirely eliminate) the impact of losses or errors on any single interface.
2. Cost : This will provide a perfect combination between cost and quality of service, as a minimum level of service can be guaranteed with lower cost, by combining several interfaces of different cost parameters.
3. Redundancy : Multiple interfaces can provide effective solution for maintaining connections for sensitive and/or time–critical applications.

Application level techniques to manage flow control, data re–assembly and traffic distribution with two levels of abstraction was investigated. The lower level will provide support for core functionalities of interface selection and co–ordination while the higher level will provide application specific functions for the particular use–case. However simultaneous use of multiple interfaces cannot be implemented at this time and the development of these techniques will be one of the central objectives of future research.

Currently, usage of multiple interfaces by switching between available interfaces during a multimedia application session is supported. The details of the available network interface(s) on a mobile device and their service types and levels, are provided on the device usage profile. The proxy utilizes this information in delivering the multime-

Network Interface	**Service Type**	**Service Level**
UMTS	Guaranteed Service	1.8Mbits/s
WLAN	Best Effort	–

Table 4.2: Network Interface Section of Sample Usage

dia streams. Two service types are defined (as shown table 4.2), 'Guaranteed Service' and 'Best Effort'. In Guaranteed Service, the proxy delivers a constant bitrate (of-course the rate must be lower than the specified service level) while in Best Effort, the proxy activates the congestion control mechanism to adaptively deliver these multimedia streams. However, this depends on the responsive signaling of the changes on these interface(s) by the client. Two methods are possible: 1.) Manual changes of interfaces by the user of the mobile device. In this scenario, the client signals the proxy of changes in interfaces. The proxy provides support for seamless session mobility across interface selection(s) by defining a control messages used by the client for triggering the re–direction of multimedia data streams to a new address and adaptation of the streaming to the service type of the interface. This is particularly the use case, within the framework of the current Internet (as at the time of this thesis, the internet runs on IPv4), for accessing multimedia streams while changing access technologies. 2.) Automatic changes of interfaces by a handoff/handover solution. In this scenario, the client relies on the appropriate signaling from a vertical handoff/handover solution. This vertical handover solution [69] was used. The proxy only adapts the streaming to the new service type on the interface, while mobile IPv6 effectively maintains seamless connectivity in user mobility, and it is specifically used in the automatic network interface selection.

4.4 Error Control

Errors are inevitable in wireless environment and therefore error control systems are required in the delivery of real–time multimedia services. However due to time critical nature of these services, forward error correction techniques are the most suitable for this purpose. This technique requires additional redundancy that increases the size of data to be transmitted. The increased data transmission bandwidth requirement, could limit the effective utilization of the transmission channel thereby eroding the usefulness of the technique. It is therefore imperative that the redundant data required in the forward error correction procedure is strictly optimized to achieve great efficiency and optimum performance. However, optimization of error correction procedure is dependent on the expected error rates and this varies according to the access technology and operating environment. However error control was not implemented in this research.

It is also possible to reduce the impact of losses due to errors on the client, by closely matching the sizes of the application level multimedia packets to the data link packet sizes. However, this significantly increases the impact of jitter and thereby introduces severe processing and memory cost for jitter handling and in addition generates excessive overhead that reduces network bandwidth utilization. Alternatively larger packet sizes increases the severity of losses and introduces greater packetization delays which is propagated along the transmission path. The impact of the variation in packet sizes, relatively depends on the data transmission rates. For very low rates (9.6 – 16kbits/s achieved with low resolutions of SQCIF and QCIF) and low packets sizes, the jitter problem is less significant though the overhead introduced by headers severly decreases effective network bandwidth utilization. For a survey of the impact of these parameters, the following Figures A1, A2 and A3 presents the graphs of the generated Transmit Rates and Packet Sizes for a defined Source Rate. This measures the impact of the overhead generated by the RTP/UDP/IP header information. Some test measurements were taken that presents the comparison of the Jitter Levels and Error Rates to Packet Sizes for a defined Source Rate. This measures the impact of the packet sizes on the error and jitter rate at four different source rates using four (SQCIF, QCIF, CIF and 600x360) resolutions.

5 System Implementation

The system is implemented in a modular structure of multi–processing components. The implementation of the design framework is presented in UML and organized into top–level logical view of the entire application solution and its environment and finely detailed development view.

5.1 Logical View

The system comprises of three sub–systems within the context of the application framework. These are Content Provider, AdaptServer and Client sub–systems. The content provider sub–system represents any system that provides/implements an interface for various multimedia services. The descriptions for this sub–system will be limited to the interface it provides as the sub-system itself is an external "entity" within this application definition. The AdaptServer sub–system requires an interface from the content provider sub–system and provides adapted multimedia services to the client sub–system.

5.2 Development View

Component diagrams The AdaptServer sub–system has four (4) components and the Client sub-system has two (2) components. The implementation platform is presented in table 5.3 and the list of all components and interfaces with brief descriptions presented in table 5.1 and 5.2 respectively.

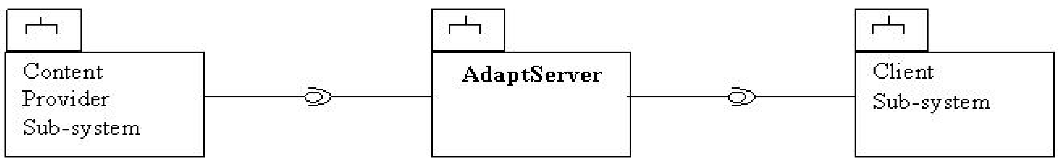

Figure 5.1: Logical View of Application Framework

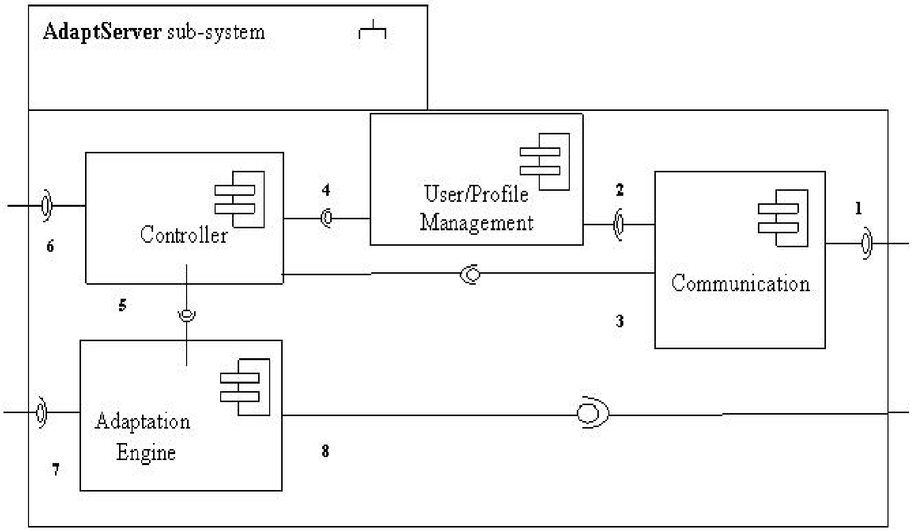

Figure 5.2: The AdaptServer Sub–system

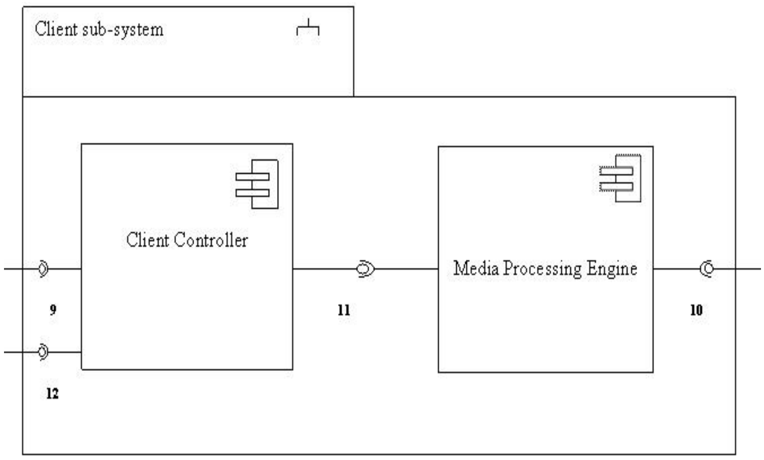

Figure 5.3: The Client Sub–system

Component	Sub–System	Description
Communication	AdaptServer	A component for the communication of control, user and profile information.
User/Profile Management	AdaptServer	Storage, processing and management of users and profile information.
Controller	AdaptServer	Controls the adaptation engine based on the control information received.
Adaptation Engine	AdaptServer	Processing and delivering of multimedia data streams based on the control information from the controller.
Client Controller	Client	Sends, receives and reacts to control information on the client.
Media Processing Engine	Client	Processes multimedia data streams.

Table 5.1: List of all components with brief descriptions

5.3 Scenarios and Use Cases

The Sequence diagrams and detailed descriptions of two selected use–cases are presented in this section. In the first use–case scenario the mobile client updates the proxy with its' usage environment description profile. This enables mobile client to access multimedia data streams through the proxy. The second use–case scenario presents the access of multimedia streams through the proxy.

1. Scenario : Profile Setup
Description : To initiate the multimedia services, this is first and non – recurring step in the entire application framework.
Preconditions : None
Sequence diagram : See Fig. 5.4
Exceptions : The defined profile response signals are: "Successful Profile Update" or "Unsupported Profile"
Post–conditions : The user's profile is registered and all subsequent service requests are delivered according to profile information.

Interface/ (Number)	**Component**	**Description**
Comm1 (1)	Communication	External interface to receive control and profile data.
Comm2 (2)	Communication/ Profile Management	Internal interface to transmit profile and user data.
Comm3 (3)	Communication/ Controller	Internal interface to transmit control information
Cont1 (4)	Profile Management/ Controller	Internal interface to transmit users' group information.
Cont2 (5)	Controller/ Adaptation Engine	Internal interface to transmit control information to the adaptation engine.
Cont3 (6)	Controller	User interface to the controller
Adapt1 (7)	Adaptation Engine	External interface to receive multimedia streams
Adapt2 (8)	Adaptation Engine	External interface to deliver processed multimedia streams.
User (9)	Client Controller	User Interface and Display for Client
Data (10)	Media Processing Engine	External interface for receiving multimedia data streams.
Control1 (11)	Client /Media Engine	Internal interface to control multimedia processing.
Control2 (12)	Client Controller	External interface for sending and receiving control information.

Table 5.2: List of the internal and external interfaces of components with descriptions

AdaptProxy Sub–System	
Platform:	Intel x86
Operating System:	Linux
Support Library:	The GStreamer Library 0.10.x [70] Sqlite database library [71]
Programming Language:	C Programming and GObject
Client Sub-System	
Platform:	ARM / Nokia 770
Operating System:	Nokia's Embedded Linux (maemo)
Support Library:	The GStreamer Library 0.10.x [70]
Programming Language:	C Programming

Table 5.3: Description of Implementation Platform

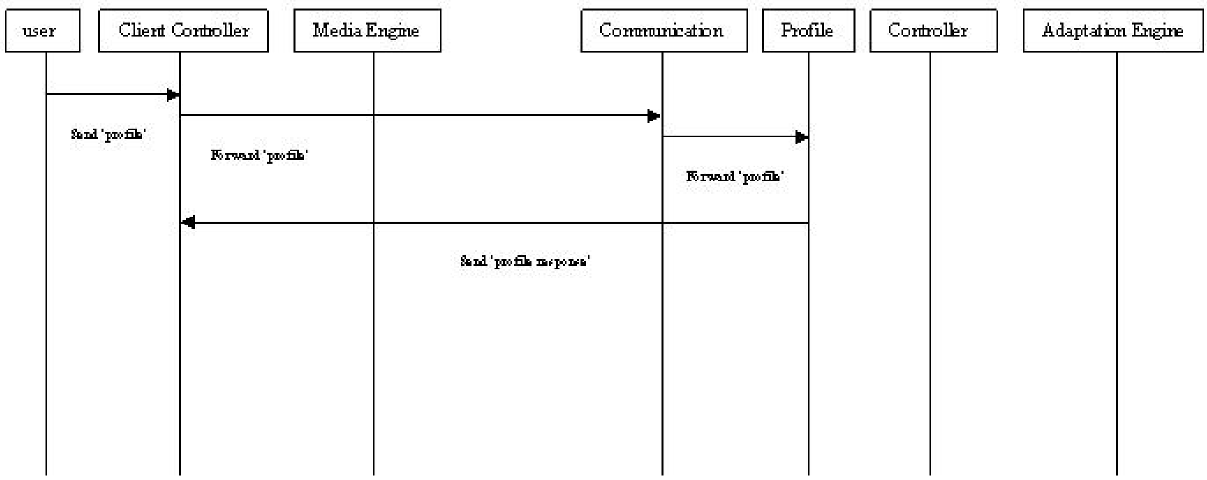

Figure 5.4: Sequence Diagram Profile Setup

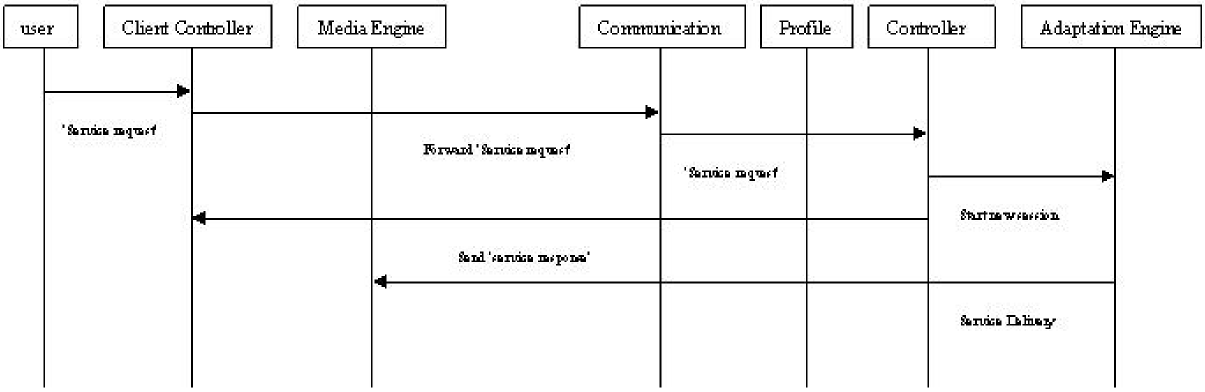

Figure 5.5: Sequence Diagram Service Initiation

2. Scenario : Service Initiation

Description : The user requests for multimedia services. The internet resource address is transmitted to AdaptServer, and thereafter receives the clients' specific adapted streams.

Preconditions : Profile Setup

Sequence diagram : See Fig. 5.5

Exceptions : Requested multimedia resource 'Not Found'

Post–conditions : The client receives control port for the adaptation engine session upon successful service initiation and delivery.

Other Use Cases

Bit rate Change : The client controller changes current bit rate due to the prevailing network and/or terminal condition, by sending the bit rate change signal to the session control port.

Session Mobility : The client address changes due to manual intervention of the user by changing the network interface and signals AdaptServer before or after the address change.

Resource Change : The client changes current internet multimedia resource, by sending resource change signal to the session control port on the AdaptServer.

6 System Test

The overall application framework was tested in several phases which includes component test and overall system test but only some selected features that define the core functionality of the entire framework is analyzed in this chapter.

6.1 Test Environment Description

The design was implemented in a three – tier structure divided into content provider, proxy and mobile client. The content provider is an independent IPTV content provider network in the Telecommunications Laboratory of the Faculty of Information Technology used for the LAILA Project, the network provides cable television and other video services to xDSL customers. The television programmes are delivered in multicast groups to the IP set–top–boxes. Each channel of the television broadcast is assigned a multicast–group. The customer joins the appropriate group (simply changes channels) to receive the preferred television programming channel. In addition other internet streaming sites and multimedia files were also used as part of content provider system. The proxy was implemented on linux machine with GStreamer multimedia library [70], and the mobile client was implemented on Nokia 770 – Nokia's Embedded Linux Mobile Device. The mobile device Nokia 770 has two network interfaces: WLAN and bluetooth (UMTS access is provided through the Bluetooth connection to a mobile phone), it was used to access streams of the IPTV broadcast, internet sites or multimedia files through the proxy. However only multimedia formats available in GStreamer is supported.

More so, two test networks were used, the first network (see fig 6.1) hosted by the LAILA project consists of the IPTV network, the proxy server and a WLAN access point. The mobile client connects through the WLAN interface and accesses television streams with the multicast addresses. The proxy provides a virtual multimedia gateway for the mobile client. the second network (see fig 6.2) is hosted by the VerHO Project. VerHO Project is joint project of the University of Jyväskylä and Jyväkylä Polytechnic. The objective of the project was to develop a mobility management system that co–ordinates the efficient and effective selection/use of multiple access networks on a mobile device. The project used the Nokia 770 mobile device for implementing a prototype for a policy based automatic network interface selection system. The test network consists

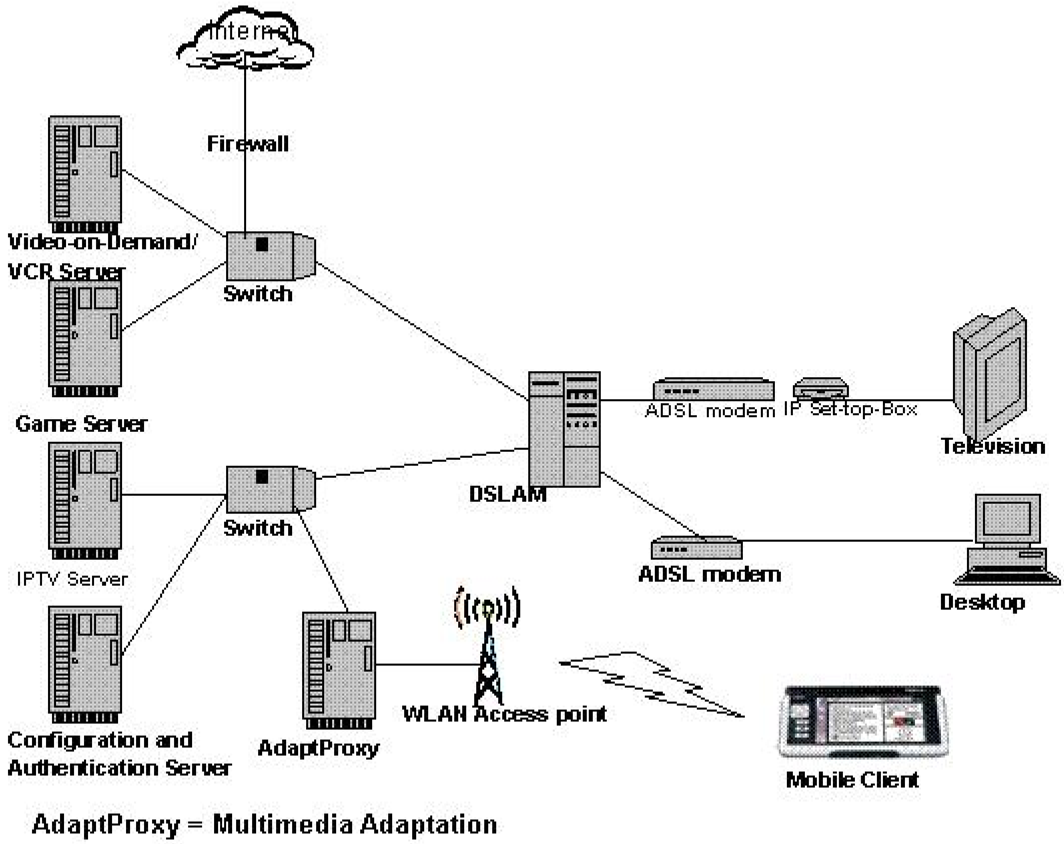

Figure 6.1: The First test network – LAILA Network: Wireless Extension of Cable Broadband Multimedia Services

of WLAN access points, UMTS connection and the proxy. The Mobile IPv6 protocol is used and the mobile device accesses multimedia files while interchanging the access networks between WLAN and UMTS through the proxy.

6.2 Test Parameters

Two test parameters are measured simultaneously in all the test scenarios presented in the next section. The parameters are :

1. Sensitivity Test: This provides a comparison of two measurements. 1.) The measurement of the proxy server data transfer rate against time and 2.) the requested data rate from the client within the same time threshold. This is used in the test scenarios to measure and analyze the sensitivity of the proxy.

2. Quality Test: This is a measurement of the Peak Signal to Noise Ratio (PSNR) against time (measured per frame basis). A video testing software VQM described in

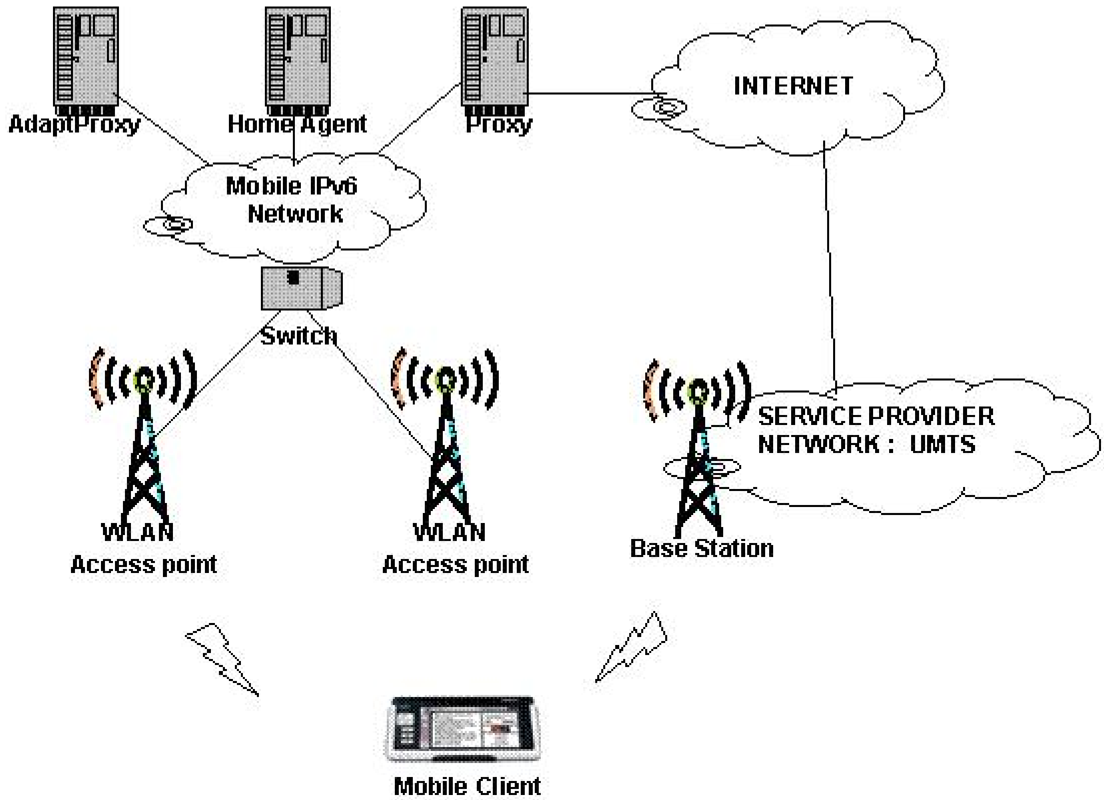

Figure 6.2: The second test network – VerHO Network: WLAN and UMTS access networks used on the mobile device with signaling of interface changes from VerHO mobility management solution

the appendix A.1 is used, and the test measures the quality of the received media at the mobile device. This analyzes the impact of the adaptation process on the overall utility of the received media.

6.3 Test Scenarios and Results

The system was tested with several scenarios which includes :

Profile Setup – setup of the mobile device profile
Service Request – Service initiation based on previous device profile, the proxy uses the device profile to initiate services
Bit–Rate Change – Dynamic change of source rate
Session Mobility – The re–direction of the multimedia streams to a new connecting address on the same mobile node

The performance of the tests were in line with expectation, the proxy delivered multimedia streams according stored device profiles and dynamic adjustment to stream source rate was supported.

However the following selected test scenarios are presented with the results from the two(2) test parameters.

1. Adaptive delivery of multimedia services over best effort network. In this scenario, the delivered multimedia services were dynamically adapted to the client rates' request. A test video sequence with the following properties was used:
Duration : 46 seconds
Framerate : 15 frames per seconds

However, the received video was broken into three(3) separate clips for the quality tests in–order to ensure temporal alignment, each of the clips has a duration of ten (10) seconds, broken into the following layout:
Clip1: 6s – 15s
Clip2: 16s – 25s
Clip3: 26s – 35s
This results in a total of 150 frames per video clip. The graphs in appendix (fig. A.4, A.5, A.6, A.7 and A.8) presents the results of the sensitivity tests for the entire dura-

Processor:	Pentium II
Speed:	800MHz
Memory:	512MB
Operating System:	Fedora Core 5
Library:	GStreamer 0.10.10

Table 6.1: Description of Test Platform

tion of the video sequence and quality tests for each clip respectively.

The measured response time for rate change request is 94ms, this result is dependent on the test platform which is presented in table 6.1. However the sensitivity test only measures this performance on the server itself. The transmission channel and the receiving client introduces additional delay which impacts on the performance perceived by the mobile client. The dynamic rate adjustment process results in some rate drop at the encoding point, the rates at the 16th, 27th and 35th seconds (in Fig. A.5) shows this effect. However the quality of the streams is at acceptable levels with lowest levels recorded at the Clip 2 period (Fig. A.7) that coincides with the lowest encoding bitrate for the video sequence.

7 Conclusion

The implementation validated the proposed solution to the problem of effective interconnection between the increasingly pervasive multimedia services and the continually divergent user environment (preference, terminal and connectivity). The Profile based multimedia adaptation method provided the platform to seamlessly deliver multimedia services to different and widely diverse clients, yet effectively obliterating the complexity of the task from the content provider. The objective of providing multimedia streams that support both the device properties and the users' access network properties was satisfied.

In addition, dynamic responses to network conditions by encoding rate adjustments provided the required guaranteed data throughput in best effort mobile networks. However, a relatively high computational cost is incurred in providing this transcoding services in real–time. Moreover the twin problems of jitter and losses due to errors increases the processing requirements, delay and network bandwidth usage. Several jitter management techniques were explored.

7.1 Future Research

Error correction techniques for real–time multimedia communication often requires addition of redundant data, however this has negative impact on wireless network bandwidth utilization. In addition, the advances in wireless systems will increase the availability of these technologies, offering different error properties. Therefore developing an access technology independent error correction technique that effectively reduces (if not completely eliminate) the need for data overhead in error recovery is a central point of future research.

8 References

[1] H. Holma and A. Toskala, *HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications*, vol. 05. Chichester, England: John Wiley Sons, 2006 ed., 2006.

[2] "The ieee 802.16 working group on broadband wireless access standards." IEEE Standards Association which is available at: (http://grouper.ieee.org/groups/802/16/).

[3] I. Burnett, R. V. de Walle, K. Hill, J. Bormans, and F. Pereira, "Mpeg–21: Goals and achievements," *IEEE MultiMedia*, vol. 01, no. 1, pp. 60–70, 2003.

[4] T. Crowley, P. Milazzo, E. Baker, H. Forsdick, and R. Tomlinson, "Mmconf: an infrastructure for building shared multimedia applications," in *CSCW '90: Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, (New York, NY, USA), pp. 329–342, ACM Press, 1990.

[5] K. Rothermel, G. Dermler, and W. Fiederer, "A communication infrastructure for multimedia applications," in *European Conference on Networks and Optical Communications, 1996 - netresearch.ics.uci.edu*, (Stuttgart, Germany), 1996.

[6] K. Asatani and Y. Maeda, "Access network architectural issues for future telecommunication networks," *Communications Magazine, IEEE*, vol. 36, pp. 110–114, 1998.

[7] A. Lazar, "Challenges in multimedia networking." Lazar, A. A., Challenges in Multimedia Networking, Proc. International Hi-Tech Forum, Osaka, Japan, Februray 1994. which is available at: (http://citeseer.ist.psu.edu/lazar94challenges.html), 1994.

[8] Z. Sahinoglu and S. in Tekinay, "On multimedia networks: self-similar traffic and network performance," *Communications Magazine, IEEE*, vol. 37, pp. 48–52, 1999.

[9] J. Liebeherr, "Multimedia networks: issues and challenges," *Computer Magazine, IEEE*, vol. 28, pp. 68–69, 1995.

[10] H. J. Stiittgen, "Network evolution and multimedia communication," *Multimedia, IEEE*, vol. 2, pp. 42–59, 1995.

[11] C. R. Lin and M. Gerla, "A distributed architecture for multimedia in dynamic wireless networks," *Global Telecommunications Conference, 1995. GLOBECOM '95., IEEE*, vol. 2, pp. 1468–1472, 1995.

[12] L. Trajkovic and S. J. Golestani, "Congestion control for multimedia services," *Network, IEEE*, vol. 6, pp. 20–26, 1992.

[13] T. Kwok, "Communications requirements of multimedia applications: apreliminary study," *Wireless Communications, 1992. Conference Proceedings., 1992 IEEE International Conference on Selected Topics in*, pp. 138–142, 1992.

[14] J. Chung and M. Claypool, "Better-behaved, better-performing multimedia networking," *Proceedings of SCS Euromedia, 2000*, p. 1, 2000.

[15] D. Dwyer, S. Ha, J.-R. Li, and V. Bharghavan, "An adaptive transport protocol for multimedia communication," *Multimedia Computing and Systems, 1998. Proceedings. IEEE International Conference on*, vol. 6, pp. 23–32, 1998.

[16] C. Krasic, K. Li, and J. Walpole, "The case for streaming multimedia with tcp," in *Interactive Distributed Multimedia Systems: 8th International Workshop,IDMS 2001, Lancaster, UK, September 4-7, 2001, Proceedings*, (Berlin, Germany), p. 213, Springer Berlin / Heidelberg, 2001.

[17] I. Rhee, V. Ozdemir, and Y. Yi, "Tear: Tcp emulation at receivers - flow control for multimedia streaming," in *TEAR: TCP emulation at receivers - flow control for multimedia streaming*, (NC , USA), p. 24, Department of Computer Science, North Carolina State University, 2000.

[18] T. Braun, "Internet protocols for multimedia communications," *Multimedia, IEEE*, vol. 4, pp. 74–82, 1997.

[19] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "Multimedia streaming via tcp: an analytic performance study," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, (New York, NY, USA), pp. 908–915, ACM Press, 2004.

[20] L.-A. Larzon, M. Degermark, and S. Pink, "Efficient use of wireless bandwidth for multimedia applications," in *Mobile Multimedia Communications, 1999. (MoMuC '99) 1999 IEEE International Workshop on*, (Los Alamitos, CA, USA), pp. 187–193, IEEE Computer Society, 1999.

[21] H. Zheng and J. Boyce, "An improved udp protocol for video transmission over internet-to-wireless networks," *Multimedia, IEEE Transactions on*, vol. 3, pp. 356–365, 2001.

[22] G. Ghinea, J. P. Thomas, and R. S. Fish, "Multimedia, network protocols and users bridging the gap," in *MULTIMEDIA '99: Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, (New York, NY, USA), pp. 473–476, ACM Press, 1999.

[23] A. Fox and E. A. Brewer, "Reducing www latency and bandwidth requirements by real-time distillation," in *Reducing WWW Latency and Bandwidth Requirements by Real-Time Distillation*, (Paris, France), Fifth International World Wide Web Conference, 1996.

[24] A. Fox, S. D. Gribble, E. A. Brewer, and E. Amir, "Adapting to network and client variability via on-demand dynamic distillation," *SIGOPS Oper. Syst. Rev.*, vol. 30, no. 5, pp. 160–170, 1996.

[25] A. Fox, S. D. Gribble, and Y. Chawathe, "Adapting to network and client variation using active proxies: Lessons and perspectives," *IEEE Personal Communications*, 1998.

[26] J. R. Smith, R. Mohan, and C.-S. Li, "Transcoding internet content for heterogeneous client devices," in *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, vol. 3, (Los Alamitos, CA, USA), pp. 599–602, IEEE Computer Society, 1998.

[27] H. Bharadvaj, A. Joshi, and S. Auephanwiriyakul, "An active transcoding proxy to support mobile web access," in *SRDS '98: Proceedings of the The 17th IEEE Symposium on Reliable Distributed Systems*, (Washington, DC, USA), p. 118, IEEE Computer Society, 1998.

[28] J. R. Smith, C.-S. Li, and R. Mohan, "Adapting multimedia internet content for universal access," *Multimedia, IEEE Transactions on*, vol. 1, no. 5, pp. 104–114, 1999.

[29] W. X. and S. H., "Adaptive reservation: a new framework for multimedia adaptation," in *Multimedia and Expo 2000. ICME 2000. 2000 IEEE International Conference on*, (Los Alamitos, CA, USA), pp. 1051–1054, IEEE Computer Society, 2000.

[30] D. S. Turaga, A. A. E. Al, C.Venkatramani1, and O. Verscheure, "Adaptive live streaming over enterprise networks," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, (Los Alamitos, CA, USA), pp. 974– 979, IEEE Computer Society, 2005.

[31] I. Burnett, R. V. de Walle, K. Hill, J. Bormans, and F. Pereira, "Mpeg-21: goals and achievements," *IEEE Multimedia*, vol. 10, pp. 60–70, 2003.

[32] B. Fuhrt, "Multimedia systems: An overview," *IEEE MultiMedia*, vol. 01, no. 1, pp. 47–59, 1994.

[33] A. Tanenbaum, *Computer Networks*, vol. 05. USA: Mc–Graw Hill, 2005 ed., 2005.

[34] "Mpeg-4: the standard for multimedia for the fixed and mobile web." ISO/IEC 14496 which is available at: (http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm).

[35] D. C. Markert, "From alpha to omega: An overview of the mpeg standards," *ACM*, pp. 1 – 6, 2002.

[36] C. Leonardo, "Impact of mpeg standards on multimedia industry," *Proceedings of the IEEE*, vol. 86, pp. 1222–1227, 1998.

[37] "Mpeg-2: Generic coding of moving pictures and associated audio information." ISO/IEC 13818 October 2000 which is available at: (http://www.chiariglione.org/mpeg/standards/mpeg-2/mpeg-2.htm).

[38] MediaLabTeliaSoneraFinland, "Mpeg-4 white paper," *Internet and Mobile Multimedia Research& Design*, pp. 1 – 6, 2004.

[39] K. Rob, "Profiles and levels in mpeg-4: Approach and overview," *Image Communication*, vol. 15, pp. 463–478, 2000.

[40] M. Burlacu and S. Kangas, "Mpeg-4 technology strategy analysis," Master's thesis, Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology, March 2003.

[41] "Mpeg liaisons." Internal Liaison within ISO/IEC JTC 1 which is available at (http://www.chiariglione.org/mpeg/liaisons.htm).

[42] S.-F. Chang, T. Sikora, and A. Puri, "Overview of the mpeg-7 standard," *IEEE Transactions on Circuits and Systems for Video technology*, vol. 11, pp. 688–695, 2001.

[43] "Mpeg 7 : Multimedia content description interface." ISO/IEC 15938 which is available at: (http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm).

[44] "Mpeg 7 interoperability." D-Lib Magazine Volume 5 Number 9 ISSN 1082-9873 : which is available at: (http://www.dlib.org/dlib/september99/hunter/09hunter.html).

[45] "Mpeg 21 : Universal multimedia access." ISO/IEC 21000-N which is available at: (http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm).

[46] A. Vetro and C. Timmerer, "Digital item adaptation: Overview of standardization and research activities," *IEEE TRANSACTIONS ON MULTIMEDIA*, vol. 07, no. 3, pp. 418–426, 2005.

[47] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, 2003.

[48] S.-F. CHANG and A. VETRO, "Video adaptation: Concepts, technologies, and open issues," *Proceedings of the IEEE*, vol. 93, pp. 148–158, 2005.

[49] J.-R. OHM, "Advances in scalable video coding," *Proceedings of the IEEE*, vol. 93, pp. 42–56, 2005.

[50] "Applications and requirements for scalable video coding." ISO/IEC JTC1/SC29/WG11 N6880, January 2005, Hong Kong, China.(http://www.chiariglione.org/mpeg/working_documents/mpeg-04/svc/requirements.zip).

[51] J. XIN, C.-W. LIN, and M.-T. SUN, "Digital video transcoding," *Proceedings of the IEEE*, vol. 93, pp. 84–97, 2005.

[52] M. Kropfberger, *Multimedia Streaming over best effort networks using multi-level adaptation and buffer smoothing algorithms.* Phd thesis, Klagenfurt University Austria, Universitat Klagenfurt. Alpen-Adria-Universitat Klagenfurt Universitatsstrabe 65-67 9020 Klagenfurt Austria, 2004.

[53] K. Nakauchi, H. Morikawa, and T. Aoyama, "A network-supported approach to layered multicast," *IEEE International Conference on Communications, 2001. ICC 2001. Helsinki, Finland*, vol. 4, pp. 1227–1231, 2001.

[54] "Multi-session rate control for layered video multicast." In Proc. of Multimedia Computing (http://www.cc.gatech.edu/fac/Mostafa.Ammar/papers/MMCN-full.ps.gz).

[55] "Stability and fairness issues in layered multicast." In Proceedings of Nossdav, 1999 (http://www.nossdav.org/1999/papers/38-1224802202.pdf).

[56] D. Rubenstein, J. Kurose, and D. Towsley, "The impact of multicast layering on network fairness," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 169–182, 2002.

[57] K. Asatani and Y. Maeda, "Access network architectural issues for future telecommunication networks," *Communications Magazine, IEEE*, vol. 36, no. 8, pp. 110–114, 1998.

[58] S. Y. Hui and K. H. Yeung, "Challenges in the migration to 4g mobile systems," *Communications Magazine, IEEE*, vol. 41, no. 12, pp. 54–59, 2003.

[59] J. D. Vriendt, P. Laine, C. Lerouge, and X. Xu, "Mobile network evolution: a revolution on the move," *Communications, IEEE*, vol. 40, no. 4, pp. 104–111, 2002.

[60] "Ieee 802.11tm wireless local area networks (wlan)." IEEE Standards Association which is available at: (http://grouper.ieee.org/groups/802/11/).

[61] "Ieee 802.20 mobile broadband wireless access (mbwa)." IEEE Standards Association which is available at: (http://grouper.ieee.org/groups/802/20/index1.html).

[62] T. Braun, "Internet protocols for multimedia communications. i. ipng-thefoundation of internet protocols," *Multimedia, IEEE*, vol. 4, no. 3, pp. 85–90, 1997.

[63] T. Braun, "Internet protocols for multimedia communications. ii. resourcereservation, transport, and application protocols," *Multimedia, IEEE*, vol. 4, no. 4, pp. 74–82, 1997.

[64] S. P. A, "Mpeg-2: a tutorial introduction to the systems layer," *IEEE Colloquim*, pp. 411–413, 1995.

[65] G. Franceschini, "The delivery layer in mpeg-4," *Image Communication*, vol. 15, pp. 347–363, 2000.

[66] H. Schulzrinne and J. Rosenberg, “The ietf internet telephony, architecture and protocols,” *IEEE Network*, vol. 07, no. 3, pp. 18–23, 1999.

[67] D. Lindbergh, “The h.324 multimedia communications standard,” *IEEE Communication Magazine*, vol. 01, no. 1, pp. 46–51, 1996.

[68] J.-C. Lapayre and F. Renard, “Appat: A new platform to perform global adaptation,” *dfma*, vol. 00, pp. 351–358, 2005.

[69] J. Puttonen, G. Fekete, J. Mäkelä, T. Hämäläinen, and J. Narikka, “"using link layer information for improving vertical handovers",” in *Personal Indoor and Mobile Radio Conference, 2005. PIMRC 2005. Proceedings of the 16th Annual IEEE International Symposium on*, vol. 3, (Berlin, Germany), pp. 599–602, IEEE Computer Society, 2005.

[70] “Gstreamer – open source multimedia framework.” GStreamer – Open Source Multimedia Framework which is available at: (http://gstreamer.freedesktop.org/).

[71] “Sqlite – sqlite embedded library.” Sqlite – Sqlite Embedded library which is available at: (http://www.sqlite.org/).

A Appendix

A.1 Video Quality Metric (VQM)

The Video Quality Metric is a robust video testing tool, supporting several test models. Details can be found at http://www.its.bldrdoc.gov/n3/video/vqmsoftware.htm

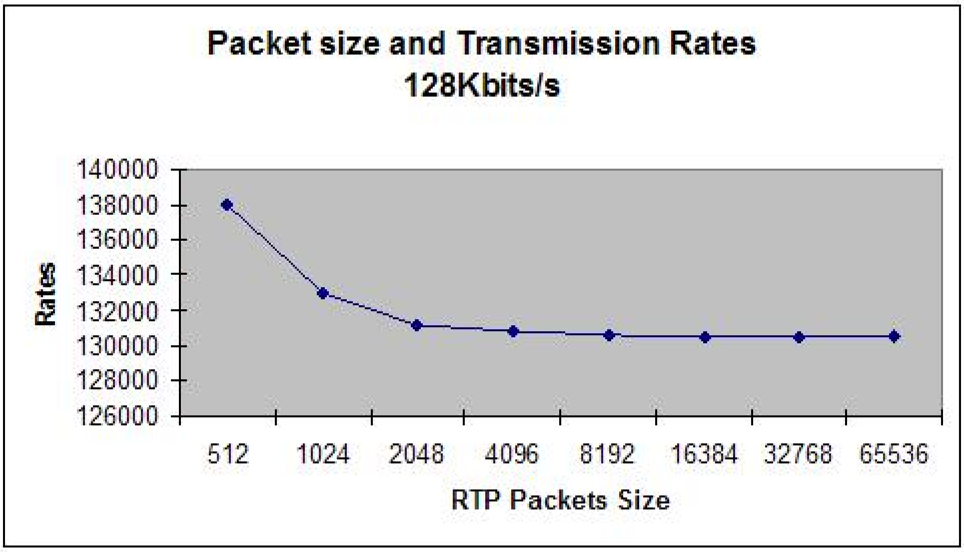

Figure A.1: Packet size and Transmission Rates at source rate of 128kbits/s

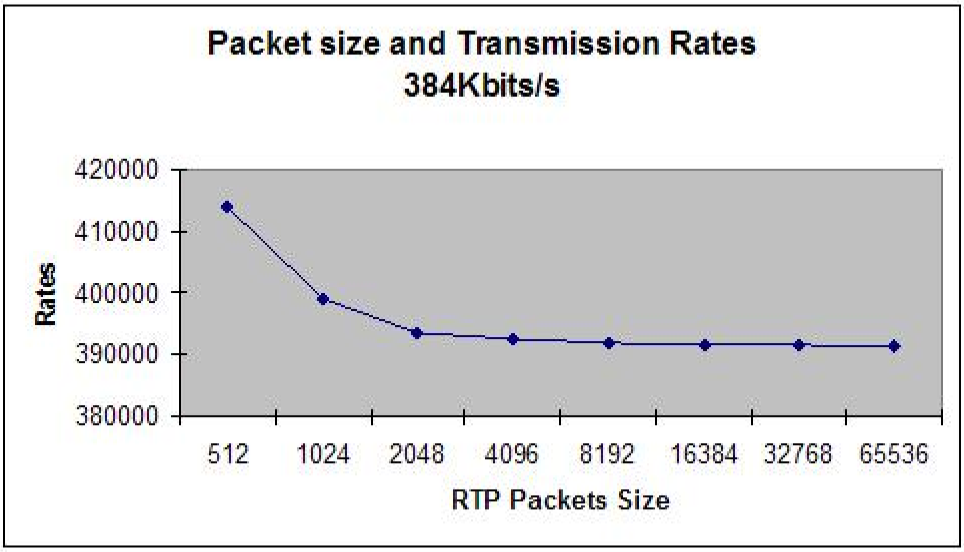

Figure A.2: Packet size and Transmission Rates at source rate of 384kbits/s

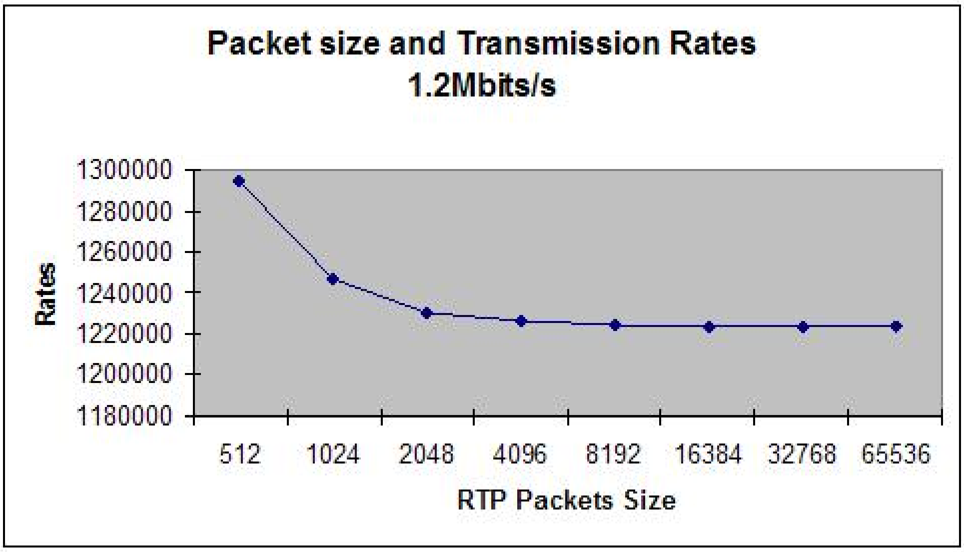

Figure A.3: Packet size and Transmission Rates at source rate of 1.2Mbits/s

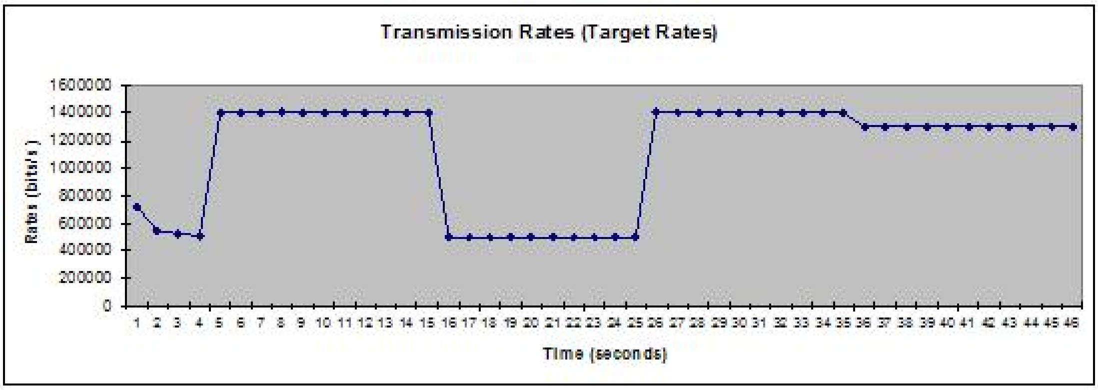

Figure A.4: Sensitivity Test: Graph of the target data transfer rate. Adaptive delivery of multimedia services over best effort network

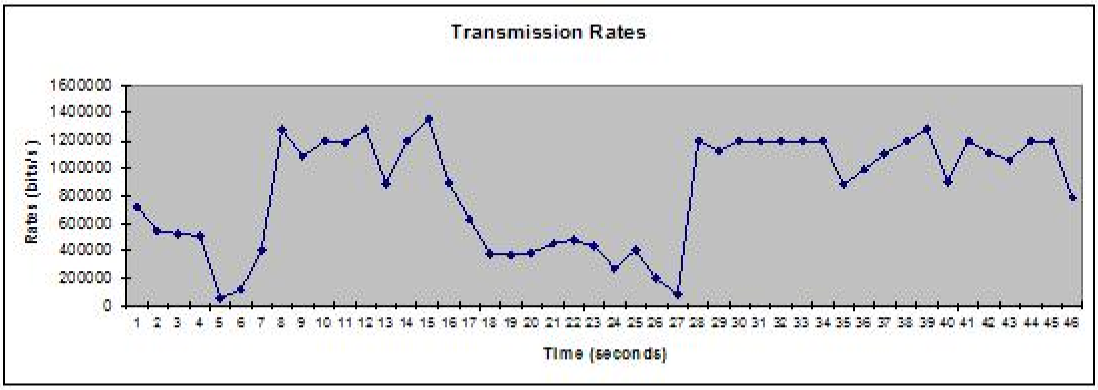

Figure A.5: Sensitivity Test: Graph of the actual data transfer rate. Adaptive delivery of multimedia services over best effort network

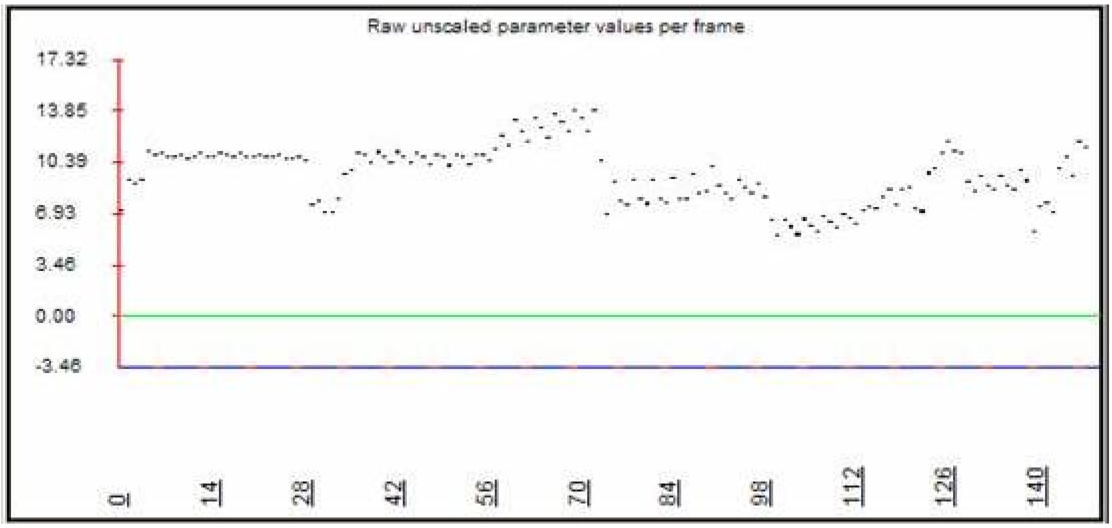

Figure A.6: Quality Test – Video Clip 1: Adaptive delivery of multimedia services over best effort network. The PSNR values uses the Double Stimulus Continuous Quality (0 to 100, where 0 is no impairment).

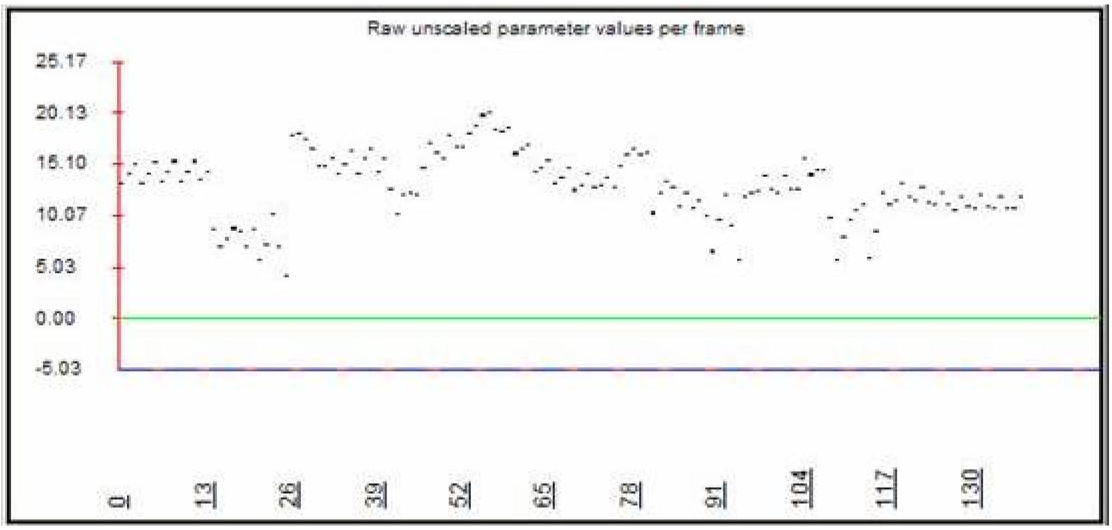

Figure A.7: Quality Test – Video Clip 2: Adaptive delivery of multimedia services over best effort network. The PSNR values uses the Double Stimulus Continuous Quality (0 to 100, where 0 is no impairment).

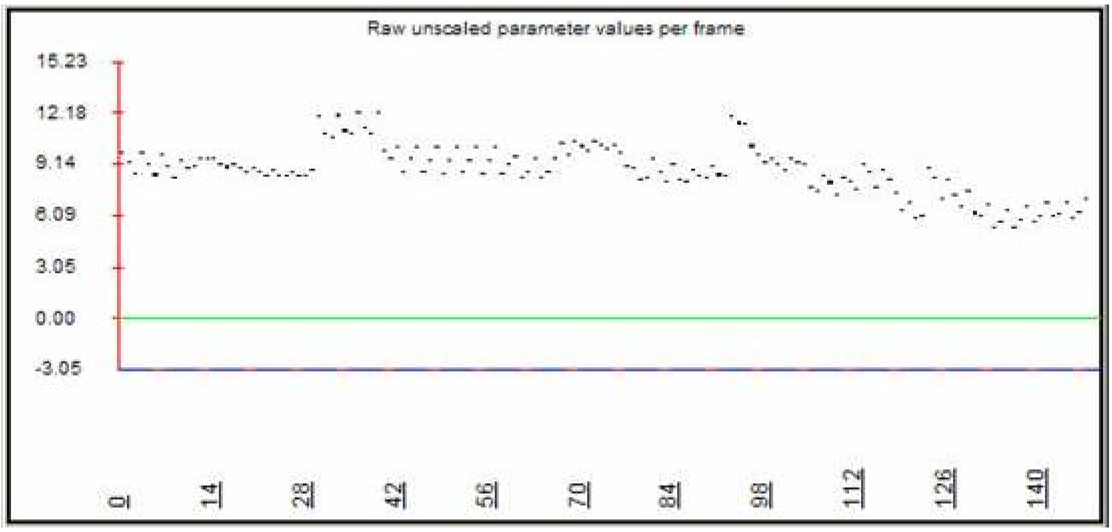

Figure A.8: Quality Test – Video Clip 3: Adaptive delivery of multimedia services over best effort network. The PSNR values uses the Double Stimulus Continuous Quality (0 to 100, where 0 is no impairment).