Adaptive Header Compression Techniques for Mobile Multimedia Networks

A.Cellatoğlu

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Centre for Communication Systems Research
School of Electronic and Physics Sciences
University of Surrey
Guildford, Surrey GU2 7XH, UK

February 2003

Summary

IP-based multimedia services generally require the use of the Real-time Transport Protocol (RTP), which is typically deployed on top of UDP/IP. RTP provides end-to-end network transport functions, which are suitable for transporting real-time applications, such as audio, video and/or data over multicast and unicast networks. However, the main drawback of using RTP/UDP/IP protocol stacks is the relatively large overhead imposed by these headers, which are at least 40/60 bytes in total, for each transmitted packet depending on whether IPv4 or IPv6 is used. Such significantly large overheads cause the transmission to be inefficient in mobile networks, due to the limited bandwidth and the behaviour of the cellular link.

This thesis investigates methods for the efficient and error resilient header compression/optimisation schemes over mobile networks. The proposed header compression schemes are implemented, and error resilient techniques are developed to render them more immune to channel variations. This includes header part-of-interest prioritisation using unequal protection with a view to increasing the robustness to channel errors. The effectiveness of these resilient techniques is demonstrated through the results of simulated VoIP and video-over-IP transmission.

Packets can have significant variations in arrival time, due to the time-varying characteristics of the networks. This will cause the packets to arrive out of order, or alternatively with a high degree of arrival-time jitter. Variation in this delay will not only make compression schemes inefficient, but also degrade the output quality of the decoding applications, and in particular, the intelligibility of speech services will be significantly affected.

In light of these facts, a new adaptive-robust and efficient header compression/stripping scheme is introduced. Within this scheme "application defined packets" and "smart packets" are introduced, which improve robustness. "Application defined time-windowing" and "smart-adaptive buffering" techniques are established, which have a significant impact on spectrum efficiency. Extensive network simulations demonstrate the effectiveness of the proposed schemes throughout the research work.

Acknowledgements

I would like to express my deep sincere gratitude to my PhD supervisor Prof. Ahmet KONDOZ for his close concern, help, guidance and support throughout my research. I am and will be very much indebted to my supervisor and also Dr. Simon FABRI, and Dr. Stewart WORRALL, for their invaluable help and encouragement at various stages of my research work. They have been my second supervisor as well as close colleagues during my PhD period and they always will.

I would like to take this opportunity to express my gratefulness to my wife Nazlı for her endless love and understanding she has shown during my PhD work. Finally, my deepest and most sincere thanks to my mother MŰNEVVER, and my father HASAN, and to my sister Aydan, to my brothers Ali and Hakan, and to rest of my dear family members, especially my aunty FEZILE, for their endless love, support, encouragement and understanding they have provided under all kinds of circumstances thought out all my life time. Last but not least, very special thanks also go to number of close friends, Khaldoon Al-Naimi, Safak DOGAN and especially Maria Farrugia, whose friendship will always be highly valued.

Contents

Chapter 1

1 Introduction

1.1 Preamble and Objectives

Recent trends in mobile telecommunications show a significant shift towards packet-switched communications, due to increasing interest in multimedia-rich content. This follows the success of Internet-based communications, which uses packet-switched technology, as well as the success of second-generation short messages (SMS) and voice only mobile communications. As a result, the drive towards extending the range of services for mobile users is set to increase the role played by multimedia technology in upcoming mobile communications systems. Hence, Third Generation (3G) networks are designed to support IP over the air interface. IP-based networks grant very high service flexibility and application independence, facilitating a multitude of real-time and interactive services. As a result, 3G networks will enable a wide range of real-time IP-based multimedia applications to operate over the mobile networks. The challenge is to provide acceptable Quality of Service (QoS) guarantees in the same way as-is done in circuit-switched networks, which provide voice services with optimal quality and spectrum efficiency.

There are a number of problems when using packet-switched technology for the delivery of multimedia services over mobile environments. One of the main problems is that wireless channels have a limited bandwidth as they are shared between many users, making the radio spectrum a costly resource in cellular links. This is an issue, because IP-based real-time multimedia services require the use of an appropriate transport protocol such as the Real-time Transport Protocols (RTP). This provides end-to-end network transport functions, which are suitable for transporting real-time applications, such as audio, video, and/or data, over multicast and unicast network services. RTP is typically deployed on top of the UDP/IP protocols. The combined RTP/UDP/IP headers have a length of at least 40 bytes. This includes the IP header (20 octets), the UDP header (8 octets) and the RTP header (12 octets). If IPv6 is used, the total overhead is increased to 60 bytes. When operating over low throughput links, or when transmitting speech or audio streams, which have been compressed to low data rates, these headers often require more bits than the payload. The headers represent a considerable proportion of the total throughput, thereby decreasing transmission efficiency.

The second important characteristic is the lossy nature of cellular links. Wireless networks usually have much higher residual bit error rate (BER) than wired networks. Since the overheads are significantly large, it is very likely that an error occurs within the header during transmission. These errors in the RTP/UDP/IP headers may cause the packet to be lost, even if there is no error in the payload. As a result of this, the packet loss rate (PLR) increases significantly and effectively affects the quality of service.

So, even though real-time multimedia communication over-IP will bring many advantages to mobile networks, the Internet transport protocols are not well suited for wireless networks. Therefore, a major challenge is to reduce/compress this excessive header information in a reliable, robust method so as to be used over relatively error prone and narrow band cellular channels. Maintaining transparency with external IP networks sets another challenge.

In summary, the objectives of the research work are to establish the basics for a generic header compression scheme that can be used over error-prone environments. The work proposes solutions to achieve the following:

- decrease overhead for multimedia transmissions
- reduce packet loss rate over lossy channels due to header corruption
- adapt to the delay variations of the core network

- allow the use of small packets to minimise the latency in delay sensitive low-rate links

The rest of the chapter is organised as follows: the second section discusses how the performance evaluations were carried out in the thesis. The third section presents the original achievements of this research whilst the fourth section focuses on the outline of the thesis.

1.2 Source Material and Performance Evaluation

The performance evaluations of the header compression algorithms presented in this thesis are divided into three categories. The first one is the packet loss rate (PLR) that is caused by header corruption. A packet loss event can occur for following two reasons:

- A bit error might damage the compressed header and prevent the de-compressor from reconstructing the original header.
- The context[†] of the de-compressor may lose synchronisation with the context of the compressor, possibly preventing subsequent packets from being decompressed. This can happen even if subsequently received headers are error free

The average PLR is calculated by using the Equation 1.1

$$Average.PLR(\%) = \frac{P_t - P_r}{P_t} \times 100 \qquad \text{Equation 1.1}$$

Where, P_t and P_r represent the number of sent packets by the compressor over the mobile link and the number of correctly re-constructed packets at the de-compressor, respectively.

The second performance evaluation method is intended to demonstrate the efficiency of the developed algorithms. The efficiency is measured by calculating the average header size used per packet as shown in Equation 1.2

$$AverageHeaderSize = \frac{\sum_{i=0}^{i=t} H_i}{P_t} \qquad \text{Equation 1.2}$$

[†] The context is the state, which is used to compress and decompress a header by the compressor and de-compressor.

Where, “i” and “t” represent the packet number and total number of packet respectively and the “H” represents the size of header that is used to transmit packet.

Finally, the speech and video quality performance evaluations of the results obtained can be measured either subjectively or objectively. Subjective tests require a number of users to view and compare a number of different decoded video sequences. Thus, subjective tests would require a group of people to spend a lot of time viewing large numbers of sequences. Clearly, this is not a convenient option, although it is the most reliable metric. Therefore, for speech quality; average Signal-to-Noise Ratio (SNR) of the particular decoded speech files is used. For the video quality; the most common metric, which is average Peak-to peak Signal to Noise Ratio (PSNR) of a particular decoded video sequence, is used. The equation for SNR and PSNR are shown below, in Equations 1.3 and 1.4.

$$SNR = 10Log_{10}\frac{\sum x^2}{\sum(x-\hat{x})^2} \qquad \text{Equation 1.3}$$

$$PSNR = 10Log_{10}\frac{255^2}{\frac{1}{M*N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(x(i,j)-\hat{x}(i,j))^2} \qquad \text{Equation 1.4}$$

Where M and N stand for the dimensions of the image and x and $\hat{x}$ are the original and decoded image data, respectively. Here, M and N correspond to 176X144 pixels in case of the Quarter Common Intermediate Format (QCIF) image quality evaluation. Table 1-1 gives a rough indication of the quality vs. PSNR relationship that one might expect from encoding QCIF sequences.

PSNR Range (dB)	Quality
< 20	Totally unacceptable, unintelligible
20 – 25	Subjective is perceptible, unacceptable quality
25 – 28	May be acceptable, degradation visible
28 – 32	Very little degradation visible
> 32	Excellent image quality

Table 1-1 Guide to PSNR values

1.3 Original Achievements

The research work presented in this thesis is aimed at the development of generic header compression algorithms with error resilience capability for the transmission of multimedia services over heterogeneous mobile networks. A number of publications have been produced as a result of the research that is described in this thesis, and papers are listed in Appendix A. In this thesis, the work, which is believed to be original contribution, can be summarised as:

- Investigation of the performance of two different header compression algorithms including proposed error resilience techniques, Compressing **RTP** (CRTP) and **RO**bust **C**heck-sum header **CO**mpression (ROCCO), over **G**eneral **P**acket **R**adio **S**ervice (GPRS) mobile channels.
- The design of an *enhanced CRTP* header compression scheme, which uses a new packet format, *compressed reference_header packet* within CRTP.
- A new error resilience technique, called *Slow Start Update Scheme (SSUS)*, used in the environments where the end-to-end delay is high and continuously varying.
- The design of an *enhanced ROCCO* header compression scheme by using two new error resilience methods, which are:
    - *Adaptive Reference Update Scheme (ARUS)* to minimise/stop the risk of context damage propagation within the ROCCO algorithm. This new error resilience technique includes a new packet format called *adaptive_reference header.*
    - *Prioritisation* of packets using different bearer channels for headers in packets with different sensitivities to errors. It is built on top of the ARUS method.
- The design of a novel adaptive header compression system called *Adaptive Time-Windowing (ATW) and Adaptive Forward-Buffering (AFB) scheme* for use by multimedia services in wireless environments. Within this scheme the following new packet formats are proposed,
    - *The Aplication_Defined Packet (ADP),* which is used to manage the adaptive forward buffer more efficiently and smooth out the variability
    - *The Smart Packet (SP),* which is used to maintain synchronisation between mobile terminal and **U**p-**L**ink **E**dge **P**roxy (ULeP) as well as the **D**own**L**ink **E**dge **P**roxy (DLeP) and the end user.

1.4 Thesis Outline

Chapter 1 contains a brief introduction to the background of the thesis and outlines the reasons behind the work. In this chapter, the objectives of the work are also defined. To meet the objectives, quality assessment techniques, which are used throughout the thesis, have been introduced. Moreover, this chapter has also presented outlines of the original contributions of the work featured in this thesis.

Chapter 2 presents the technological background for this thesis. It discusses current and future mobile communications technologies. It presents the evolution of networks including their capability for supporting multimedia services in wireless channels. This allows the identification of the main requirements of header compression algorithms associated with providing multimedia services over cellular links.

Chapter 3 introduces header compression technologies for use over mobile links. The advantages that can be gained by using header compression are clearly defined. CRTP is implemented, and evaluated over simulated GPRS mobile channels. The error resilience techniques, *TWICE* and *periodic refreshes* are described and evaluated. In addition, new technologies the ***compressed reference_header packet format*** and the ***slow start update scheme (SSUS)*** are proposed and evaluated. The performance of these techniques is compared with standard CRTP and with other error resilience techniques interims of robustness and efficiency.

Chapter 4 focuses on a header compression approach gaining support in the mobile communications industry, called ROCCO (***RO***obust *C*hecksum-based header *CO*mpression), which was initially proposed in 1999. Initially, this scheme is implemented and simulated over mobile environments. In this chapter, two new resilience techniques are proposed to improve the performance of the ROCCO. The first one, the **Adaptive *Reference_Update Scheme (ARUS)*** helps to minimise/stop the risk of context damage propagation within the ROCCO algorithm. This new error resilience technique includes a new packet format, called the ***adaptive_reference header***. The second resilience method is called **Prioritisation**, which is built on top of the ARUS method. These are tested over mobile channels to measure the packet loss rate and efficiency performance. They are evaluated by using various standard speech codecs, which have different characteristics, and the MPEG-4 video codec, to observe the effect on the speech and video quality.

Chapter 5 presents an alternative to standard header compression architectures known as, the **"Adaptive Time-Windowing (ATW) and Adaptive Forward-Buffering (AFB) scheme"**, which addresses problems associated with CRTP and ROCCO, as well as the characteristics of end-to-end communications using packet-switched technology. This system minimises header sizes thereby conserving bandwidth and radio spectrum, and also minimises the affect of the varying transmission delay. It is designed to provide smooth play-out of packets over the air interface on the downlink and minimises the computational complexity and power requirements on the handset by providing 0-byte header compression. Also, it is designed to ensure that consecutive packets are independent of each other's in wireless environments preventing error propagation found in CRTP and ROCCO schemes. One of the essential requirements of this scheme is to palace a **Reverse Proxy** (RP) at the Edge of the Core Network (EoCN) between end-user terminal and CN. Within this scheme, an "**application-defined packet (ADP)**" which is used to manage the adaptive forward buffer more efficiently and smooth out the variability and a "**smart packet**" are proposed, which is keep synchronisation between mobile terminal and ULeP as well as the downlink edge proxy and the end user. This is achieved by notifying the receiving ends when the silence period is finished for speech and when the new frame is started for video.

The Chapter 6 contains the overall conclusions for the thesis, and possible directions for future research in the area.

Chapter 2

2 Mobile Packet Networks and Multimedia Technologies

2.1 Introduction

The rapidly increasing interest in mobile multimedia communications has been significantly driven by the success of the Internet communications which is based on the packet-switched technology, as well as the success of bandwidth limited second generation mobile communications. Today, there are several separate, vertically oriented, single-service networks (see Figure 2-1) [28], which have been optimised to deliver fixed telephone (PSTN), mobile telephone (GSM), limited data, and cable-TV services. These, except data services, are connection-oriented, where the end-to-end resources are reserved for the whole duration of the connection. The networks of those services are based on circuit-switched communication, which is characterised by a limited available bandwidth. They are designed to carry real-time voice traffic between hundreds of millions of phones all around the globe, see Figure 2-2 [32]. By contrast, the data communications typically refer to non-real-time applications, known as connectionless services as well, such as email, fax, Internet browsing, etc., and are based on the packet-switched technology. The service quality of data communications is based on a best-effort approach. The packet-switched technology does not set any restriction on the application, it is an application independent technology, and because of that, Internet applications have grown and spread extensively.

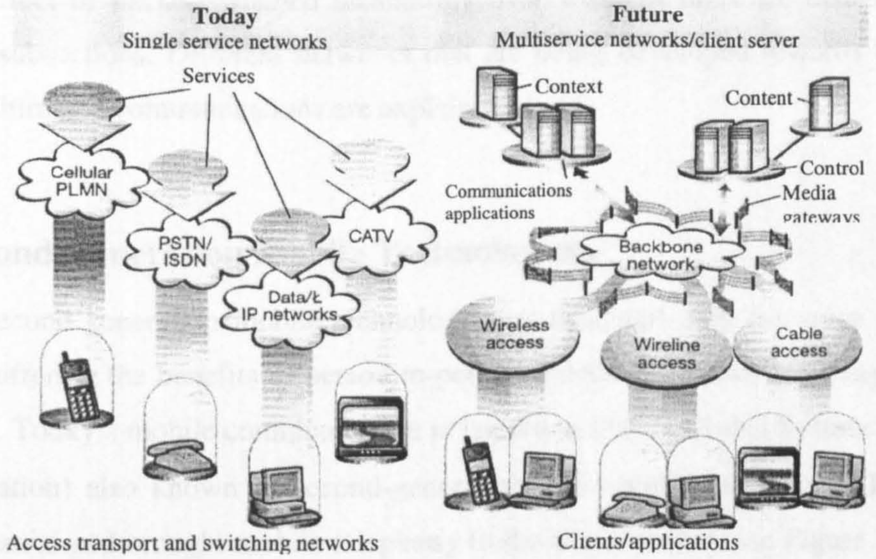


Figure 2-1: Towards next-generation multiservice networks (source [28])

This widespread growth of the Internet has opened a new era for mobile multimedia and the information technologies. As a result, the drive towards extending the range of services for mobile users, are set to increase the role played by multimedia technology in future mobile communication systems. Certainly, mobile communications are no longer restricted to voice-only and short messages (SMS) data services. Mobile technology promises to deliver real-time multimedia services over wireless links that cannot be avoided. Wide ranges of different services are all being promised to transport at the same time and in real-time to the end mobile users. These services are expected to include high-quality audio, speech, and video, computer-generated graphics and animations, together with a whole host of applications and applets which will be used to integrate all these media components together so as to provide new value-added services. So, delivering Multimedia applications over wireless link is the main challenge of the future technology. The requirements of mobile multimedia communication systems place two major demands upon the performance of underlying networks. In the case of the transmission of *bandwidth demanding error-sensitive applications*, such as video transmission, high levels of subjective visual quality are achieved only at relatively high data rates, while real-time interactive applications such as voice communication, and also videoconferencing require strict *end-to-end delay* of service guaranties.

In this chapter, current and future communication technologies with enabling services are introduced. The requirements for mobile multimedia communications, as well as the negative effect of packet-switched technology over wireless links are discussed in the following subsections. Different networks that are being developed towards the goal of mobile multimedia communications are explained.

2.2 Second Generation Mobile Technologies

Initially, second generation mobile technology was designed only for voice application purposes, offering the benefits of person-to-person speech communication anywhere and at anytime. Today's mobile communication is known as GSM (Global System for Mobile communication) also known as second-generation radio-mobile telephony. It has been very successful and brought mobile telephony to the mass market, see Figure 2-2 [32]. It is based on circuit-switched networks, and more details are given below.

2.2.1 GSM

GSM (Global System for Mobile communication) [5] is a digital mobile telephone system that is widely used in Europe and other parts of the world. It uses a variation of time division multiple access (TDMA). It is operating in either the 900 MHz or 1800 MHz frequency band in Europe. Specially equipped GSM terminals can connect with PSTN, ISDN, Packet Switched and Circuit Switched Public Data Networks, through several possible methods, using synchronous or asynchronous transmission.

GSM was designed having interoperability with ISDN in mind, and the services provided by GSM are a subset of the standard ISDN services. The services offered by GSM system are restricted to conversational narrowband services, which make speech services the most basic and most important teleservice. The various data services are supported, with user bit rates up to 9600 bps [5] [39], which is not sufficient for real-time multimedia services, because higher date rates require for multimedia applications, especially video transmission. Other GSM services apart from the interactive speech application include a cell broadcast service, where messages such as traffic reports, are broadcast to users in particular cells. A service unique to GSM, the Short Message Service, allows users to send and receive point-to-point alphanumeric messages up to a few tens of bytes. It is similar to paging services, but much more comprehensive, allowing bi-directional messages, store-and-forward delivery, and acknowledgement of successful delivery.

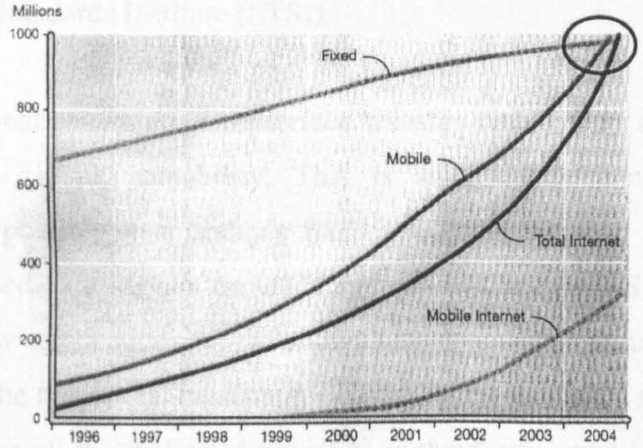


Figure 2-2: The growth of fixed and mobile telephony and Internet (source [32])

Unfortunately, the mobile communications are not voice and short data generic anymore. The current growth of personal communications is also driven by the success of the Internet Protocol [22] [27] [33] [37]. IP flexibility and reliability has generated a growing interest in packet-switched technology over wireless links. As shown in Figure 2-2 [32], the strong growth of mobile communication is expected to continue based on the "Erricson" research, in 2001. The real-time multimedia service expectations over wireless links are also rapidly increasing as well [58]. However, GSM capabilities are clearly not sufficient for the provision of multimedia services, including simultaneous video and audio transmission. Initially, to overcome this problem, and expand services over wireless networks, new technology has been introduced which is packet-based wireless communication that can run over GSM system. These are known as General Packet Radio Services (GPRS), which are explained in detail in the following section.

2.2.2 GPRS

The General Packet Radio Service (GPRS) [3] [21] [55] is a new non-voice value added service that allows information to be sent and received across a mobile telephone network. It is an end-to-end mobile packet communication system, which makes use of the same radio architecture as GSM [5]. In order to achieve such coexistence, GPRS introduces the Packet Data Traffic Channels (PDTCHs) [123], which are used to transfer the user information. GPRS is also the name for an international packet-switched networking standard in GSM systems, initially developed by the European Telecommunication Standards Institute (ETSI).

GPRS offers theoretical maximum air-interface transfer rates of up to 171.2 kb/s [21] with its multimedia slotting capability. This is significantly faster than the data transmission speeds possible over today's fixed telecommunication networks and the current circuit-switched data service on GSM networks. Thus, GPRS promises to fully enable the use of new applications on the move with increased communication speeds. However, achieving the theoretical maximum GPRS data transmission speed requires that a single user take over all the dedicated timeslots without any error protection. Clearly, since it is unlikely that a network operator will allow all timeslots to be used by a single GPRS user, transfer rates lower than 171.2 kbit/s are more likely to be pronounced in realistic systems subject to the mobile terminal capabilities and carrier interference. Moreover, GPRS facilitate instant connections whereby information can be sent or received immediately as the need arises, subject to the radio coverage.

Coding Scheme	*Convolutional Code Rate*	*Payload per Block [bits]*	*User Bit Rate [kbit/s]*
CS-1	1/2	181	9.05
CS-2	~2/3	268	13.4
CS-3	~3/4	312	15.6
CS-4	1	428	21.4

Table 2-1: GPRS channel coding Scheme

GPRS involves overlaying a packet-based air interface on the existing circuit-switched GSM network. This gives the user an option to use a packet-oriented data service. Therefore, with the use of GPRS, the information is split into separate but related packets before being transmitted and re-assembled at the receiving end. User data packets are segmented, coded and transformed into radio blocks. Each radio block is further interleaved over four standard GSM normal bursts and transported across the air interface in the same manner as the circuit-switched speech is transmitted in GSM. When an error occurs in the transmission medium, data packets can be re-transmitted at the radio block level.

A new set of logical channels has been defined for GPRS traffic as opposed to the circuit-switched networks where all the signalling and information transfers make use of one channel only. This set includes control channels and packet data traffic channels. A physical channel allocated for GPRS traffic is called a packet data channel (PDCH). The PDCH consisits of a multi-frame pattern that runs on timeslots assigned to GPRS [20] [21]. Thus, the GPRS data is transmitted over the PDCH and is protected by four different channel schemes: CS-1, CS-2, CS-3 and CS-4 [5]. The channel coding is used to protect the transmitted data packets against transmission errors. CS-1-to-3 use convolutional codes and block check sequences of varying strengths so as to produce different rates. CS1-3 are based on a 1/2 rate convolutional code, which is punctured to obtain approximate rates 1/2, 2/3 and 3/4, respectively. On the other hand, CS-4 is uncoded whereby it only provides error detection functionality [141]. Each of the four channel protection schemes is assigned a maximum of eight timeslots [3]. The coding schemes and resulting bit rates per one timeslot are described in Table 2-1.

Delay Class	*Delay (maximum values)*			
	SDU size: 128 octets		*SDU size: 1024 octets*	
	Mean Transfer Delay (sec)	*95 percentile Delay (sec)*	*Mean Transfer Delay (sec)*	*95 Percentile Delay (sec)*
1. Predictive	*< 0.5*	*< 1.5*	*< 2*	*< 7*
2. Predictive	*< 50*	*< 25*	*< 15*	*< 75*
3. Predictive	*< 50*	*< 250*	*< 75*	*< 375*
4. Best Effort	*Unspecified*			

Table 2-2: GPRS Service Classes

The choice of one of the four coding schemes for the coding of PDCHs depends on the quality of the channel and also on the application's QoS requirements. Under very bad conditions, a very reliable CS-1 may be used and a data rate of 9.05 kbit/s per GPRS timeslot can be obtained. Under good condition, data can be transmitted without convolutional coding and a transport rate of 21.4 kbit/s per timeslot can be achieved. Consequently, with the use of eight slots of this scheme, namely CS-4, a maximum data rate of 171.2 kbit/s can be obtained in theory. However, in practice, multiple users may sometimes share the timeslots resulting in a much lower bit rate for an individual user [21] [142].

Although it is expected that GPRS will have a considerable impact on the types of data services that can be offered over a mobile network, it has two major limitations. First, as it employs the same access architecture as GSM systems, the throughput of GPRS access channels is limited to the amount of information bits that can be transmitted using a GMSK carrier at a system rate of 270 ksym/second. Secondly, GPRS currently offers only very poor end-to-end delay guarantees, as can be seen from delay and reliability service classes shown in Table 2-2 (55). These limitations are being addressed, both directly and indirectly, in the standardisation work being carried out in the development of E-GPRS, Enhanced Data Rates for GSM Evolution (EDGE) [86] 87], which is one step closer to the 3G network, UMTS. These networks are explained in the following sub-section.

2.3 Third Generation Technologies

After the success of the short-message and voice only second-generation GSM mobile applications, the promised services to the mobile end users have been expanded. However the expanded services are not supported by GSM system due to the limitation from many aspects, such as bandwidth, usage of channel, etc. On the other hand Internet usage growth is increasing rapidly and statistic results (see Figure 2-2) show that it will continue to increase, without any application limitations. So, this does not leave any other option than the marriage of the mobile communication and IP, which can bring new era to the mobile communications. Therefore, in future, third-generation radio-access technology promises to introduce new technology, a multiservice network, which extends beyond today's basic telephony, (see Figure 2-1) [28] and provide an extended service to the mobile users. The terminals used in the mobile multimedia era will nearly always be connected to the network, serving as the gateway to the Internet or to corporate intranets via packet-switched connectivity networks. This will eliminate delays associated with connection set-up, and add convenience to the use of data and multimedia services. [40]

In the direction of 3G-communication technology, the second-generation communication has been upgraded by GPRS, which is the first network that introduced packet-switched technology to the mobile user. So, the next generation of data heading towards third generation and personal multimedia environments builds on GPRS. Although, GPRS has delivered some new services to the mobile environments, none of the GPRS service classes shown in Table 2-2 [3] [21] provide for quality of service levels sufficient for the provision of low-latency packet speech services. These limitations are being addressed, both directly and indirectly, in the standardisation work being carried out in the development of E-GPRS, Enhanced Data Rates for GSM Evolution (EDGE) [86], which has been standardized in ETSI for Release 99 in 1999, and it was referred to as EDGE phrase-1 and later in the year EDGE phase-2 was released and it has been called GERAN, which is counted as a third-generation technology.

2.3.1 GERAN (E-GPRS)

The E-GPRS (GERAN = GSM-EDGE Radio Access Network) standard is built on the existing GSM standard, using the same existing cell arrangements, and 200 kHz radio access network, as GPRS. However, E-GPRS introduces a new modulation and coding scheme to facilitate an increase in the throughput capacity of GSM systems. In order to retain compatibility with current GSM GMSK (Gaussian Minimum Shift Keying) systems, the symbol rate of 270 ksym/sec is retained, but a 3-bit/symbol 8-PSK-modulation scheme is employed. It is designed to deliver data at rates up to 384 Kbps by increasing the radio data rate per timeslot from 22.8 kbps to 59.2 kbps [21], virtually a three-times increase, and enable the delivery of multimedia and other broadband applications to mobile phone and computer users. These schemes employ both GMSK and 8-PSK modulation methods so as to be able to provide optimum throughput in a wide variety of C/I conditions. In practice, a gain in throughput is obtained with 8-PSK only with high C/I conditions experienced by about 50 % of users in a typical urban cell [143]. EGPRS is based on the same concept as GPRS, and indeed, most of the specifications regarding the core network and much of the radio network remain unchanged. However, the different transmission characteristics of 8-PSK (Phase Shift Keying) modulated signals require the use of much more efficient and rapidly acting link adaptation techniques, together with different block structures to accommodate the necessary modulations in the protocols. One of the main advantages of GERAN is that it supports 4 main QoS classes, which are listed below by adequate radio access bearers (RAB). The radio access bearers (RAB) are used for different services through the radio access network. These radio bearers specify the required service quality and supply information on the characteristic of the traffic flow.

- The conversational service class is used for real-time services, such as ordinary telephony voice – that is IP-telephony and video-conferencing. The vital characteristics of this class are low transmission delay and preserved time relationships, or low-delay variation, in the traffic flow.

- The streaming service class applies to real-rime audio and video streaming applications. In contrast to the conversational class, this category consists of one-way transport.
- Typically associated with the interactive service class is Internet browsing and telnet. The fundamental characteristic of this service class is a request-response pattern. Therefore, the round-trip delay is an important factor
- The background service class is used for best-effort traffic. Examples of services in this class are electronic mail (e-mail), short messages service (SMS), and file transfer. In this class, the requirements that apply to transfer delay are less strict.

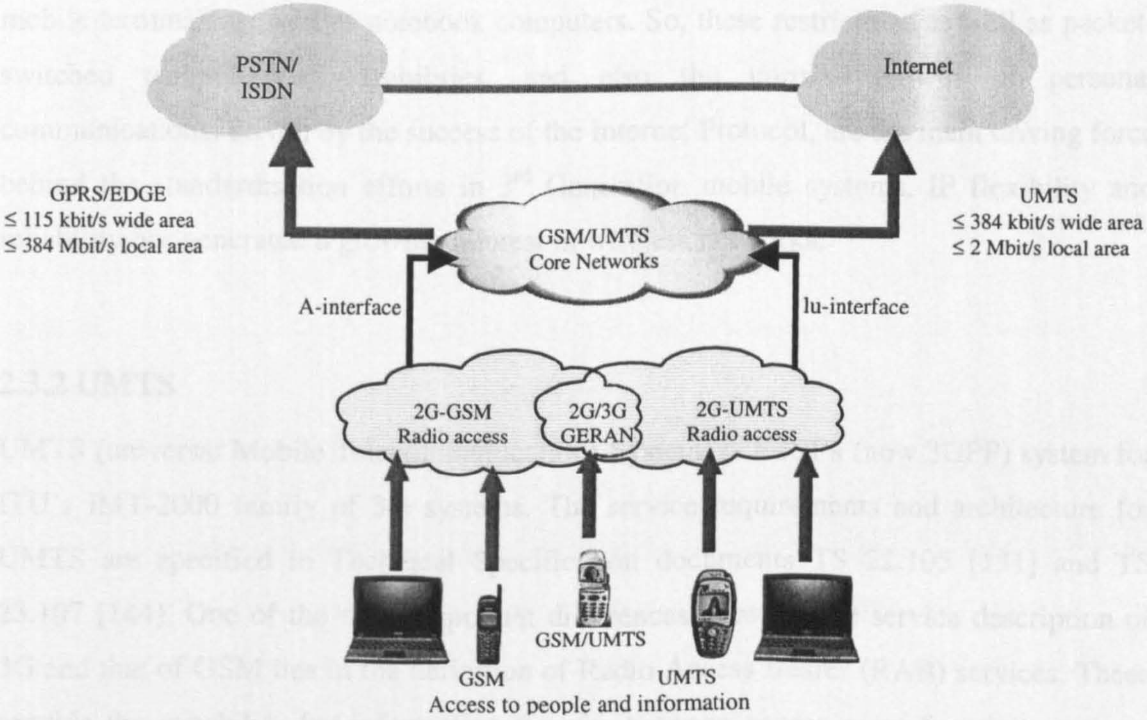


Figure 2-3: Network evolution between GSM and Third-generation mobile multimedia communication (UMTS) and GERAN position in the technology

The other advantages of the GERAN, which are very important for the mobile operators and vendors, the prime driving forces, are directly related to the low operating costs, which increase the revenues. This is because it only requires relatively small changes to network hardware and software on the current GSM networks, as it uses the same TDMA

(Time Division Multiple Access) frame structure, logic channel and 200 kHz carrier bandwidth as today's GSM networks.

Since GPRS/EDGE is run on the GSM system, it would be advantageous if their radio access network is able to support real-time voice services in packet-switched environments, however, access to the network is only available in circuit-switched mode, making it particularly inefficient for services such as email access, video conferencing and web browsing which have bursty traffic characteristics. Operating in packet-switched mode at the radio interface increases the radio capacity by implementing statistical multiplexing techniques. However, this improved rate is still does not sufficient to allow users to have continuous Internet connections on their mobile phones and notebook computers. The higher data rates are required to allow users to take part in videoconferences and interact with multimedia Web sites and similar applications using mobile terminals as well as notebook computers. So, these restrictions as well as packet-switched technology's capabilities, and also the current growth of personal communications, driven by the success of the Internet Protocol, are the main driving force behind the standardisation efforts in 3rd Generation mobile systems. IP flexibility and reliability has generated a growing interest in wireless networks.

2.3.2 UMTS

UMTS (universal Mobile Telecommunications System) is ETSI's (now 3GPP) system for ITU's IMT-2000 family of 3G systems. The service requirements and architecture for UMTS are specified in Technical Specification documents TS 22.105 [131] and TS 23.107 [144]. One of the most important differences between the service description of 3G and that of GSM lies in the definition of Radio Access Bearer (RAB) services. These provide the capability for information transfer between access point functions and are characterised by a set of end-to-end characteristics with certain Quality of Service requirements. This contrasts with the traditional use of teleservices in GSM such as end-to-end speech telephony or the Short Message Service (SMS) in which the service, or end-application is defined on an end-to-end basis.

Error Tolerant	Conversational Voice and Video	Voice Messaging	Streaming audio and video	Fax
Error Intolerant	Telnet, Interactive games	E-commerce, WWW browsing	FTP, still image, paging	E-mail arrival notification
	Conversational (delay <<1 sec)	Interactive (delay approx. 1 sec)	Streaming (delay<10 sec)	Background (delay >10 sec)

Table 2-3: 3GPP Services and Characteristics – from [131]

This decoupling of the application layer from the bearer services allows for a far greater range of services to be offered over UMTS than has previously been possible over current mobile networks. In order to allow greatest service flexibility, when negotiating the characteristic of a bearer service both the network's transfer capabilities must be defined as well as the information quality characteristics. These are defined in terms of the maximum transfer delay, the delay variation, the bit error ratio and the data rate between two access points in a given period of time. The relationship between the services to be offered and their corresponding QoS requirements are shown in Table 2-3. Current 3GPP services will be divided into interactive and distribution services. Interactive services will include real-time bi-directional conversational services such as video conferencing, and messaging services where combined audio, video, text and data messages may be forwarded to the user's mailbox. Retrieval services will also be allowed thereby enabling the user to access information from a multimedia information centre. The secondary category of services can be visualised as a form of teleservice-type service, which is, broadcast in particular channels and accessible channels and is accessible to all subscribed users.

It is immediately obvious that the service characteristics described in Table 2-3 are very similar to the high-end Internet applications currently available. The Internet is the only global communication infrastructure allowing access to a wide range of information archives, with increasing availability of access to multimedia services, such as video conferencing and streaming video. This was acknowledged in [25], where Internet access is seen as being the main use of UMTS data services, and given the ever-increasing dominance of multimedia content within the Internet, support will include distributed 3D gaming, multiparty conferencing in virtual rooms, telepresence, location-sensitive information services, and Internet TV and radio services.

	Real Time (Constant Delay)	**Non Real Time (Variable Delay)**
Operating environment	**BER/Max Transfer Delay**	**BER/Max Transfer Delay**
Satellite (Terminal relative speed to ground up to 1000 km/h for plane)	Max Transfer Delay less than 400 ms BER 1e-3 – 1e-7 (Note 1)	Max Transfer Delay less than 1200 ms or more (Note 2) BER 1e-5 - 1e-8
Rural outdoor (Terminal relative speed to ground up to 500 km/h) (Note 3)	Max Transfer Delay 20-300 ms BER 1e-3 – 1e-7 (Note 1)	Max Transfer Delay 150 ms or more (Note 2) BER 1e-5 – 1e-8
Urban/Suburban outdoor (Terminal relative speed to ground up to 120 km/h)	Max Transfer Delay 20-300 ms BER 1e-3 - 1e-7 (Note 1)	Max Transfer Delay 150 ms or more (Note 2) BER 1e-5 – 1e-8
Indoor/ Low range outdoor (Terminal relative speed to ground up to 10 km/h)	Max Transfer Delay 20-300 ms BER 1e-3 - 1e-7 (Note 1)	Max Transfer Delay 150 ms or more (Note 2) BER 1e-5 - 1e-8
NOTE 1: There is likely to be a compromise between BER and delay NOTE 2: The Max Transfer Delay should be here regarded as the target value for 95% of the data NOTE 3: The value of 500 km/h as the maximum speed to be supported in the rural outdoor environment was selected in order to provide service on high speed vehicles (e.g. trains). This is not meant to be typical value for this environment (250 km/h is more typical)		

Table 2-4: 3GPP QoS Requirements – from [131]

In order to be able to support these demanding applications, UMTS will have to provide much higher bit rates than are provided on current systems. The 3GPP Services and Systems Aspects Technical Specifications document [131] requires that a single terminal may have access to a total of 144 kbit/s in a satellite radio or rural environments, up to 384 kbit/s in urban radio environments and 2048 kbit/s in indoor and low-range pico-cells. In addition, QoS classes for UMTS are divided into two major categories. Non-real time services will have to meet very stringent BER requirements, with error rates as low as 10^{-8}, while real-time, constant delay systems may experience errors at rates as high as 10^{-3}. The QoS requirements in different reception environments are given in Table 2-4.

2.4 Multimedia Communication Internet Protocols (IP)

The packet-switched network entity's behaviour is governed by the control programs called network or Internet protocols. These Internet (IP) protocols were first developed in the late-1970s, by the Defence Advanced Research Projects Agency (DARPA). The Internet Protocol (IP) is the method or protocol by which data is originally sent from one computer to another on the Internet with packet-switched technology. The idea behind it was to establish communication with heterogeneous connectivity, without establishing a specific end-to-end channel. However, the challenge is to implement end-to-end multimedia mobile services based on the IP.

With circuit-switched technology, necessary resources (such as radio bearer channels) are allocated by network to the specific service per user at a time for the duration of the call. In contrast, the packet-switched technology is a connectionless protocol which means that there is no continuing connection between the end points that are communicating. This technology runs on a hierarchical protocol stack, which provides the end-to-end connectivity (i.e. addressing, routing) and quality of service (i.e. congestion control, error control). There are a number of protocol stacks that act as 'glue' that binds the Internet together. A simplified version of this is shown in Figure 2.4. As can be seen in Figure. 2.4, each network layer employs different protocol stacks, such as RTP, UDP, TCP, RTSP, SDP etc., according to the applications and different purposes. The layer common to all applications is the network layer, which encapsulate Internet Protocol (IP) stacks. The other protocol stacks are encapsulated on top of IP. Combination of the application data and the protocol stacks is called a packet. Then, each packet is transmitted over the IP core network, and is sent to the other end without pre-allocating any channel for transmission. They are treated as an independent unit of data without any relation to any other unit of data. In the following sub-sections, the basic elements and operations of these key Internet protocols are briefly explained.

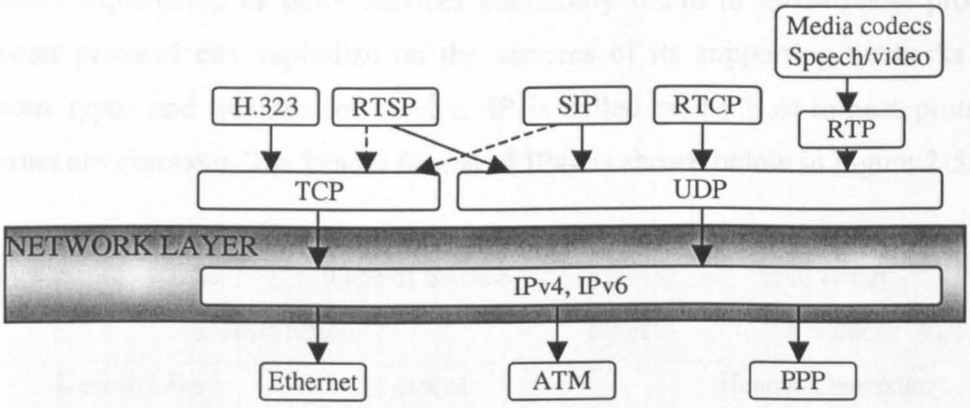


Figure 2-4: Simplified IP Protocol Stacks Architecture

2.4.1 Network Layer

The Network Layer is responsible for establishing a network wide connection between two transport layer protocol entities. It includes such functionality as network routing (addressing) and flow control across the terminal-to-network interface. In the case of internetworking it provides various harmonizing functions between the inter-connected networks. These various functions together are called the IP protocol stack. To date, there are two different versions of IP protocol stacks that are employed by the network layer. The current widely used version number is 4 and is referred to as ***IP version 4***, or, simply ***IPv4***. The second version number is 6 and is referred to as ***IP version 6***, or, simply ***IPv6***. Both of them are explained in the following sections.

2.4.1.1 IPv4

The Internet Protocol version 4 (IPv4) [8] is a network-layer protocol that contains addressing information and some control information that enables packets to be routed. IP has two primary responsibilities: providing connectionless, best-effort delivery of datagrams (a package of bits) through an Internet network, and providing fragmentation and reassembly of datagrams to support data links with different maximum-transmission unit (MTU) sizes.

The Internet protocol is specifically limited in scope to provide the functions necessary to deliver an Internet datagram from a source to a destination over an interconnected system of networks. There are no mechanisms to augment end-to-end data reliability, flow control, sequencing, or other services commonly found in host-to-host protocols. The Internet protocol can capitalize on the services of its supporting networks to provide various types and qualities of service. IP is called on by host-to-host protocols in an Internet environment. The header format of IPv4 is shown below in Figure 2-5.

<table>
<tr><td>Ver</td><td>IHL</td><td>Type of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td>Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source Address</td></tr>
<tr><td colspan="5">Destination Address</td></tr>
</table>

Figure 2-5: IPv4 Format

Version: 4 bits

Indicates the version of IP currently used.

IP Header Length (IHL): 4 bits

Internet Header Length is the length of the Internet header in 32 bit words, and thus points to the beginning of the data.

Type of Service: 8 bits

The Type of Service provides an indication of the abstract parameters of the quality of service desired. These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through a particular network.

Total Length: 16 bits

Total Length is the length of the datagram, measured in octets, including Internet header and data.

Identification: 16 bits

An identifying value assigned by the sender to aid in assembling the fragments of a datagram.

Flags: 3 bits

Various control Flags.

Fragment Offset: 13 bits

This field indicates where in the datagram this fragment belongs.

Time to Live: 8 bits

This field indicates the maximum time the datagram is allowed to remain in the Internet system.

Protocol: 8 bits

This field indicates the next level protocol used in the data portion of the Internet datagram.

Header Checksum: 16 bits

It is a checksum on the header only.

Source Address: 32 bits

It is the IP source address.

Destination Address: 32 bits

It is the IP destination address.

2.4.1.2 IPv6 - Next Generation Protocol

IPv6 [11] is a new version of the Internet Protocol based on IPv4. With the explosion of interest in Internet technology, it is likely that in the future, it will be used for many more applications and scenarios, especially in the networks where users require different services with different QoS levels. It became apparent that IP had to evolve and become more flexible. So, IPv6 has been introduced which would solve a variety of problems that would be faced with IPv4. Its major goals are;

- Support billions of hosts, even with inefficient address space allocation
- Reduce the size of the routing tables
- Simplify the protocol, to allow routers to process packets faster
- Provide better security than IPv4
- Pay more attention to the type of service, particularly for real-time data
- Make it possible for a host to roam without changing its address
- Allow the protocol to evolve in the future
- Permit the old and new protocols to coexist for years

A significant advantage of IPv6 is that it increases the IP source and destination address size from 32 bits to 128 bits, to support more levels of addressing hierarchy, a much greater number of addressable nodes and simpler auto-configuration. Scalability of multicast addresses is also introduced. The second major improvement of IPv6 is the simplification of the header. It contains only seven fields. This change allows routers to process packets faster and thus improve throughput. The third major improvement is the change in the way IP header options are encoded, allowing more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future. The last major improvement is, flow labelling capability. A new capability has been added to enable the labelling of packets belonging to particular traffic flows for which the sender requests special handling, such as non-default QoS or real-time service. The format of IPv6 is shown below in Figure 2-6

<table>
<tr><td>Version</td><td>Priority</td><td colspan="3">Flow Label</td></tr>
<tr><td colspan="3">Payload Length</td><td>Next Header</td><td>Hop Limit</td></tr>
<tr><td colspan="5">Source Address (16 bytes)</td></tr>
<tr><td colspan="5">Destination Address (16 bytes)</td></tr>
</table>

Figure 2-6: IPv6 Format

Version: 4 bits

Indicates the version of IP currently used.

Priority: 8 bits

Enables a source to identify the desired traffic class of the packets.

Flow Label: 20 bits

Used by the source to label those packets for which it requests special handling by the IPv6 router. The flow is uniquely identified by the combination of a source address and a non-zero flow label.

Payload Length: 16 bits

Total Length is the length of the datagram, measured in octets.

Next Header: 8 bits

Identifies the type of header immediately following the IPv6 header

Hop Limit: 8 bits

8-bit integer that is decremented (by one) by each node that forwards the packet. The packet is discarded if The Hop Limit is decremented to zero.

2.4.2 Transport Layer

The Transport Layer acts as the interface between the higher application-oriented layers and underlying network dependent protocol layers. It provides the layer above with a message transfer functionality that is independent of the underlying network layer. By providing the layer above with a defined set of message transfer facilities the transport layer hides the detailed operation of the underlying network from the layer above. The transport layer offers a number of classes of service to compensate for the varying quality of services (QoS) provided by the network layers associated with different types of network. The most commonly used transport protocols are User Datagram Protocol, (UDP), and Real-time Transport Protocol (RTP) for real-time applications and Transmission Control Protocol (TCP), which is used for non-real-time services, and is known as a connection-oriented protocol. The definitions of these protocols are provided with their format in the following subsections.

2.4.2.1 UDP

User Datagram Protocol (UDP) [10] is the simple best-effort transport protocol that adds multiplexing and optional checksum to IP. It is known as an unreliable protocol. It provides a mechanism for sending data over IP using very little additional overhead [10]. Because the protocol is very simple, it does not introduce any significant delay, unlike TCP. UDP is commonly used for real-time voice/video communication due to their sensitivity to delay. Its main disadvantage is that it provides no facilities for ensuring that the data arrives at the receiver. Also, there is no method of determining whether packets arrive in the correct order either. This is an issue because packets can be lost due to congestion in the network and also due to characteristics of the wireless link (e.g. high bit error rate).

UDP has properties that make it suitable for real-time applications:

- The data rate is defined by the sending application.
- Incoming packets are delivered immediately to the receiving application, even if they arrive out of order.
- Lost packets will not cause retransmissions by the transport layer.

- For validation purpose, the UDP checksum can verify the UDP headers and the data payload.

The format of UDP protocol headers is shown in Figure 2-7:

Source Port	Destination Port
Length	Checksum
Datagram	

Figure 2-7: UDP Format

Source Port: 16 bits

It indicates the port of the sending process, and may be assumed to be the port to which a reply should be addressed in the absence of any other information.

Destination Port: 16 bits

It indicates the port of the receiving process (the port number of the destination address).

UDP Length: 16 bits

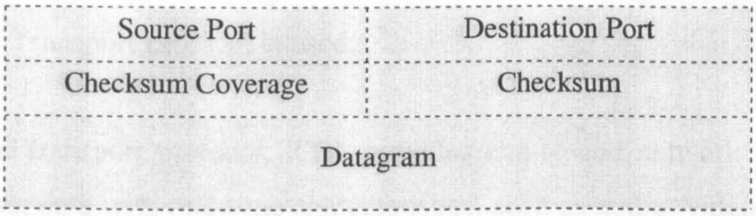
It includes its payload length, which covers RTP and the datagram.

Checksum: 16 bits

It is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the datagram.

2.4.2.2 UDP Lite

The UDP Lite [107] protocol is exactly the same as the classic UDP protocol, except it provides a flexible checksum. They are both used to meet low delay requirements. They have no overhead for retransmission of erroneous packets, in-order delivery or error correction. UDP Lite is particularly useful for real-time multimedia applications sent over links with high bit-error rates, such as mobile environments. The main reason for this is, many voice and video codes have integrated error resilience techniques, which make them significantly robust against error prone channels. Therefore, any packet with corrupted application data should not be discarded by lower layers and should be transmitted to the application layer.



Source Port	Destination Port
Checksum Coverage	Checksum
Datagram	

Figure 2-8: UDP-Lite Format

As can be seen from the UDP Lite format in Figure 2-8, the only difference between the classic UDP and UDP Lite is, the UDP Length field in the classic UDP protocol has been replaced with a Checksum Coverage field in UDP Lite. The information about the UDP Lite packet length can be found in the length field of the IP pseudo-header. The fields “Source Port” and “Destination Port” are defined as in sec 2.4.2.

Checksum Coverage:

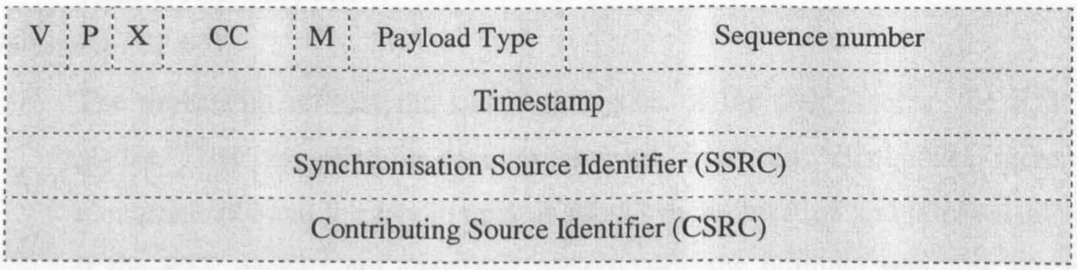
Number of bytes, counting from the first byte of the UDP Lite header, that are covered by the checksum.

The main advantage in using UDP Lite instead of the classic UDP transport protocol is that packet error rates for real-time applications are expected to decrease in wireless environments, because any error in the payload cannot fail the checksum. UDP Lite provides a partial checksum, which increases the flexibility of classic UDP by making it possible to define a packet as partially insensitive to bit errors on a per-packet basis.

2.4.2.3 RTP

One of the main weaknesses of IP for the transport of real-time services is that it only offers a best-effort service class. Packets are transmitted independently from each other, and since there is no pre-allocated channel, the packets do not necessarily follow the same route. Although, the packets are transmitted to the same destination point, because of the varying condition of the network and each route, each packet can be diverted though different routes within the core network. This can cause packets to arrive at the destination point with varying delay. This means that packets may not only be lost on the way to the destination, but may arrive out of sequence. In order to allow a receiving media player to reconstruct an original media stream from a sequence of received packets, the Real-time Transport Protocol is used.

The Real-time transport protocol, RTP, provides end-to-end network functions suitable for transmitting data with real-time characteristics, such as audio, video over multicast or unicast network services. This protocol provides functions such as timestamping and sequence numbering to facilitate the reconstruction of received media streams. RTP does not require resource reservation, but at the same time does not guarantee quality-of-service for real-time services. RTP supports multicast distribution if this is allowed by the underlying network. Therefore, in recent years the Real-time Transport Protocol, RTP [12], has been widely used for streaming audio/video transport in IP-based environments. Another reason for this success is the flexibility of RTP, which provides mutability to different application scenarios, and efficient adaptation to different network conditions. Applications typically run RTP on top of UDP to make use of its multiplexing and checksum services; both protocols contribute to the overall transport protocol functionality.



V	P	X	CC	M	Payload Type	Sequence number
Timestamp						
Synchronisation Source Identifier (SSRC)						
Contributing Source Identifier (CSRC)						

Figure 2-9: RTP Format

The RTP data packet comprises a fixed header, followed by an optional header extension and application data. The format of the fixed RTP header is shown in Figure 2-9. The first twelve octets are present in every RTP packet, while the list of CSRC identifiers is present only when inserted by a mixer. The meanings of the fields are [12];

Version (v): 2 bits

It identifies the version of RTP.

Padding (P): 1 bit

If the padding 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 MUST be followed by exactly one header extension.

CSRC count (CC): 4 bits

The CSRC count contains the number of CSRC identifiers that follows the fixed header. This number is more than one if the payload of the RTP packet contains data from several sources.

Marker (M): 1 bit

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. The receiver may use it to detect packet loss and to restore packet sequence. The initial value is randomly set.

Timestamp: 32 bits

The timestamp reflects the sampling instant of the first octet in the RTP data packet. The sampling instant is derived from the clock that increments monotonically and linearly in time to allow synchronisation and jitter calculations. If the RTP packets are generated periodically, the nominal sampling instant as determined from the sampling clock is to be used, not a reading of the system clock.

As an example, for fixed-rate audio the timestamp clock would likely increment by one for each sampling period. If an audio application reads blocks covering 160 sampling periods from the input device, the timestamp would be increased by 160 for each such block, regardless of whether the block is transmitted in a packet or dropped as silent. The initial value of the timestamp is random, as for the sequence number. Several consecutive RTP packets will have equal timestamps if they are generated at once (e.g. they belong to the same video frame).

SSRC: 32 bits

The synchronisation source (SSRC) identifier field identifies the synchronisation source.

CSRC list: 32 bits each, 0 to 15 items

The contributing source (CSRC) identifier list identifies the contributing source for the payload contained in this packet. The CC field gives the number of identifiers.

Since RTP does not support retransmission, for quality of service purposes, RTP provides another protocol, called real-time transport control protocol, RTCP. It is explained below.

2.4.2.4 RTCP

The Real-time Transport Control Protocol (RTCP) [12] is a separate reporting and control protocol that is provided by RTP. It is used to allow monitoring of the transmitted data for QoS purposes. It provides information on distribution quality (especially loss rate), and provides a uniform time reference for synchronisation between separate RTP sessions. An RTP session is limited by a preset bandwidth limit, which is the 'session bandwidth'. Within this session there can be a number of data sources, each taking up a certain amount of the session bandwidth. Generally, at the beginning of an RTP session a certain percentage of the session bandwidth is allocated to the transmission of RTCP data. This percentage may typically be in the range of 1-10%, and it works by periodic rate controlled multicast from each participant. The interval between transmissions of control packets from each participant is at a minimum once every 5 seconds, randomised up to 50% in either direction. RTP and RTCP are designed to be independent of the underlying transport and network layers

There are five different RTCP packet types. The two most common types are the Sender Report (SR) and Receiver Report (RR) packets. These are sent by RTP session participants depending on the actions of each participant. A participant sending data always sends SR packets, while a participant that is just receiving sends a RR packet. However, the format of both packet types is such that the following information will always be sent back:

- Fraction of packets lost since the last report
- Number of packets lost during the session
- Inter-arrival jitter

2.4.2.5 TCP

The Transmission Control Protocol (TCP) [7] [137] [140] is intended for use with IP, to provide a reliable method of sending data [7]. It is another transport layer protocol like UDP, but it is a connection-oriented protocol, unlike UDP. TCP connections are full duplex, which means traffic can go in both directions at same time. So, reliability is achieved through the use of positive acknowledgements, which are sent by the client when a packet is correctly received from the server. When a packet is transmitted, the server starts a timer. If acknowledgement has not been received before the timer finishes, the packet is assumed to be lost, and is retransmitted. It should be noted that this service is only reliable in terms of ensuring packet arrival, and correct ordering of packets at the receiver. An obvious disadvantage of this scheme is the delay caused by packet loss. This makes it unsuitable for real-time communications.

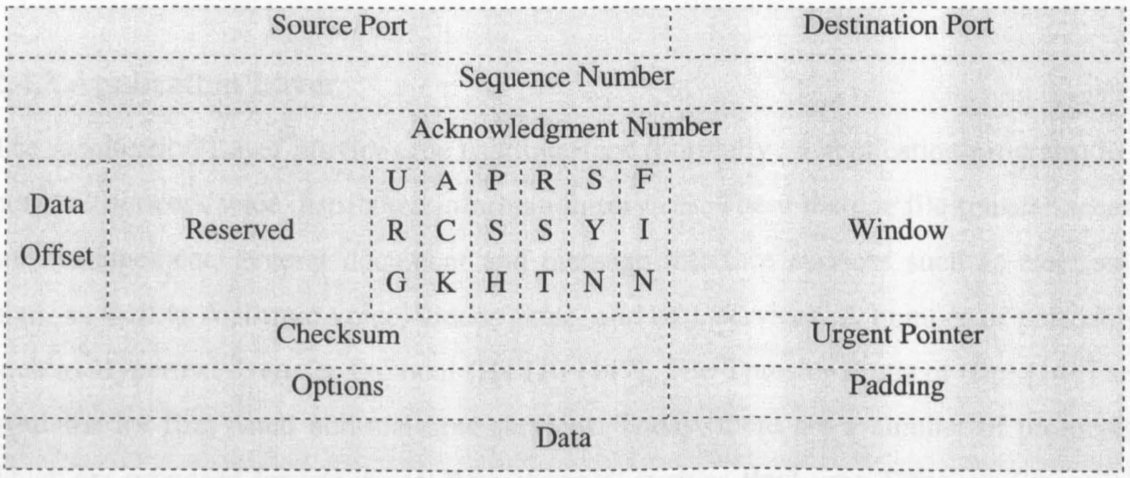


Figure 2-10: Structure of TCP header

Figure 2-10 shows the structure of a TCP header. Sequence numbers are also employed to ensure that packets are read in the correct order by the client (note the *Sequence Number* field). Each data byte effectively has its own sequence number. The value placed in the TCP header is the sequence number of the first data byte in the packet. The TCP header also contains an *'Acknowledgement Number'*, which is the sequence number of the next byte that the sender is waiting for acknowledgement of. Another field for error detection purposes is the *Checksum*, which provides error detection facilities. It is produced using the TCP header, the payload, and certain fields from the IP header.

An interesting property of the TCP header, with respect to QoS, is the ability to flag data as 'urgent'. This is achieved by setting the *URG* field to '1'. The value in '*Urgent Pointer*' then gives the offset from the beginning of the packet of the first non-urgent byte. However, the QoS offered to urgent packets is not specified, and so can vary from network to network. Without specified QoS constraints, it is difficult to efficiently send multimedia. Therefore, this is not a property that can be exploited for transmission of video over IP.

TCP headers are lengthy, and provide little functionality that is useful for delivery of multimedia. Even for streaming applications, where retransmissions are feasible, the poor level of control over retransmission makes TCP a non-ideal protocol. TCP does not support multicast or broadcast services either. The features of RTP in particular (described above) are much more attractive for multimedia applications.

2.4.3 Application Layer

The Application Layer provides the user interface (normally an application program) to a range of network wide distributed information services. These include file transfer access and management, general document and message interface services such as electronic mail, as well as real-time voice, video, game, and etc., services. A number of protocols, such as Hypertext Transfer Protocol (HTTP) [147], File Transfer Protocol (ftp) [146] are available for first listed non-real-time services. Today, there are a number of protocols, which are proposed for use in real-time services, such as Real-time Transport Streaming Protocol (RTSP) [148], Session Description Protocol (SDP) [150], and Resource reSerVation Protocol (RSVP) [151]. These are still undergoing research.

2.4.3.1 RTSP

The Real Time Streaming Protocol (RTSP) [148] is one of the application-level protocols, proposed for in 3G Technology. It is established to control either a single or several time-synchronized streams of continuous real-time media such as audio and video. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both live data feeds and stored clips. This protocol is intended to control multiple data delivery sessions, provide a means for choosing delivery channels such as UDP, multicast UDP and TCP, and provide a means for choosing delivery mechanisms based upon RTP.

It does not typically deliver the continuous streams itself, although interleaving of the continuous media stream with the control stream is possible [148]. In other words, RTSP acts as a "network remote control" for multimedia servers.

One of the main advantages of this protocol is that there is no notion of an RTSP connection; instead, a server maintains a session labelled by an identifier. An RTSP session is in no way tied to a transport-level connection such as a TCP connection. During an RTSP session, an RTSP client may open and close many reliable transport connections to the server to issue RTSP requests. Alternatively, it may use a connectionless transport protocol such as UDP.

The streams controlled by RTSP may use RTP [12], but the operation of RTSP does not depend on the transport mechanism used to carry continuous media.

2.4.3.2 SDP

The Session Description Protocol (SDP) [150] is a protocol that supports IP multicast applications in 3G. It is used purely to describe a format for conveying descriptive information about multimedia sessions. This information includes session name and purpose, session time, type of media (voice and video), media format (e.g., MPEG), transport protocol and port number, bandwidth requirements, and contact information. SDP is not a transport protocol, but relies instead on the Session Initiation Protocol (SIP) [111] [121] [145], which is explained in section 2.7.2, to deliver the session information to destinations. So, SDP is used for describing multimedia sessions for purposes of session announcement, session invitation, and other forms of multimedia session initiation. The SDP is intended to use different transport protocols as appropriate including the Session Announcement Protocol [149], Session Initiation Protocol [115] [125] [145], Real-Time Streaming Protocol [148], and the Hypertext Transport Protocol.

2.5 Mobile Link Capacity and Packet Communication

To date, packet-switched technology cannot be left out in mobile communications environments. This is because of the limitation of circuit-switched technology. Although second-generation networks provide voice services with optimised quality and spectrum efficiency, they have already reached their limit for mobile communication. Simultaneously, the Internet is based on packet-switched networks and its growth is as strong as mobile communication, but using packet-switched technology does not set any restriction with only voice services. Therefore this widespread growth of the Internet makes the packet-switched technology to become a dominating transport technology over mobile networks as well. So, to provide multimedia communication over wireless channels, packet-switched technology has become essential rather than optional. This merger brings significant advantages to wireless communications, as well as some challenges that must be tackled.

Real-time multimedia applications, such as interactive communications, voice and video, etc., are by definition, delay-sensitive services. In general, much research is based on real-time voice and video services over-IP for wireless environments. In this section, the requirements of third generation wireless services, Voice-over-IP (VoIP) [14] [32] [37] [71] and a comparison with today's voice services, are given. The voice services of third-generation wireless systems must offer at least the same high level of voice quality, and be as spectrum-efficient, as present-day second-generation systems. The challenge is to implement end-to-end IP-based transport over wireless links.

In the following sections the advantages of the packet-switched technology over mobile links, as well as the problems that arise with packet-switched technology in mobile environments, which need to be considered immediately are listed.

2.5.1 Flexibility

Service flexibility over mobile links is the most significant and needed deliverable that packet-switched technology can provide. Packet-switched communications technologies do not require the link to be dedicated between end-to-end communication points for transmission of each packet. The implication is that packets belonging to the same source may be sent via different routes, whilst other packets belonging to different sources may be sent using same path.

The main advantage of running IP end-to-end over the air interface is service flexibility, as illustrated by Figure 2-11. To date, cellular-access networks have been optimised in two-dimensional space whose X-axis and Y-axis are respectively voice quality and spectrum efficiency (see Figure 2-11). Now, the demand is service flexibility, where there are no dependencies between an application and the access network, almost anyone can develop new applications. So, a third dimension is being added in the form of IP service flexibility. By bridging the radio interface with IP packets, the services suffer a lot of protocol overhead. Therefore, for services like voice over IP over wireless (VoIPoW), the main challenge is to achieve quality and spectrum efficiency.

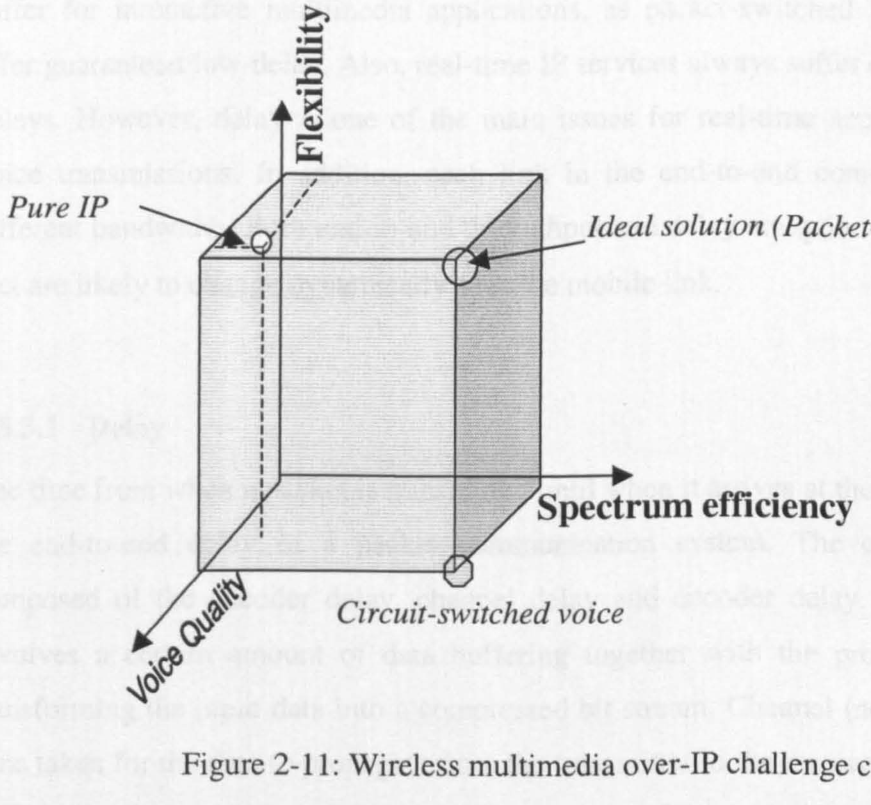


Figure 2-11: Wireless multimedia over-IP challenge cube

2.5.2 Cost

The other issue is cost. With circuit-switched technology channel pre-allocation is required by the network for the specific service for the duration of the call and users are charged based on time. Whereas, with packet-switched technology, each packet is transmitted over the IP core network, and packets are transmitted over the wireless link without pre-allocating a channel for transmission. Therefore, in theory, packet-based services are expected to cost less than circuit-switched services since communication channels are being used on a shared basis rather than dedicated to only one user at a time for a specific service. This means that the users could be charged based on the transmitted data rather than time. For example, during a speech conversation approximately 60 % of the time is silence, and only 40 % of the time is actual speech. According to these figures, packet-switched technology can offer a cheaper service than today. However, there are currently a number of issues that prevent this being true in practice.

2.5.3 Reliability

Reliability is one of the most important concerns that packet-switched technology may suffer for interactive multimedia applications, as packet-switched IP networks do not offer guaranteed low-delay. Also, real-time IP services always suffer due to unpredictable delays. However, delay is one of the main issues for real-time applications, especially voice transmissions. In addition, each link in the end-to-end connection may have a different bandwidth. Both end-to-end throughput and delay are quite unpredictable and in fact are likely to change dynamically over the mobile link.

2.5.3.1 Delay

The time from when a packet is transmitted until when it arrives at the receiver constitutes the end-to-end delay of a packet communication system. The end-to-end delay is composed of the encoder delay, channel delay and decoder delay. The encoder delay involves a certain amount of data buffering together with the processing needed for transforming the input data into a compressed bit stream. Channel (network) delay is the time taken for the data to propagate from the transmitter to the receiver whilst the decoder delay depends on the decoding processing time and the sequential arrival of the packets. Among these, core-network delay is the most critical one, since it is not predictable. Time delays greater than 0.5 seconds are usually perceived as annoying for two-way communications.

2.5.3.2 Bandwidth

Bandwidth is one of the most expensive resources for communications. Therefore, bandwidth is one of the main concerns for packet-switched applications over wireless channel. This is because the packets employ significantly large overhead that is provided by the protocol stacks. Since mobile channels have limited bandwidth and erroneous channels, using these protocol stacks is not practical, and can cause packets to be more fragile against bit errors. This is because the packet can be lost or discarded by the link layer due the corruption of the header information, while the actual application payload might be error free. This means that these overheads do not make use of a significant amount of BW resource, and that they are fragile against bit errors. In this thesis, the real-time protocol stacks are examined and different compression schemes are investigated to minimise headers in mobile channels.

2.5.3.3 Error Resilience

When transmitting information in a packet-based mobile environment, two main sources of error occur. Packet loss caused by header corruption, and delay due to congestion in the core switching network, result in the occasional loss of large, contiguous sections of the information bitsream, while fading, interference and multipath effects in the mobile channel result in individual bit errors in the received signal. As mentioned above, one of the main concerns for packet-switched applications in mobile environments is a significantly large header. Reducing the size of headers in mobile channels requires a lossless compression algorithm, which is the main concept of this thesis. The compression algorithm inevitably results in a reduction of the redundancy present in the header bitstreams, a process, which has the undesirable side effect of increasing the susceptibility of the header streams to errors. In addition, when any compression algorithm is employed, consecutive packets become dependent on each other. Therefore, any single corrupted compressed header becomes more precious as the error can propagate and can affect the following packets. Therefore, the compression algorithm should be robust enough to wireless environments and consequently prevent the propagation of errors throughout the packet sequence.

2.5.4 Complexity

Complexity is defined by the number of arithmetic computations carried out during the encoding and decoding process. The computational load in the encoder and decoder depends on the particular application. The complexity issue is also related to the power consumption. For battery life purposes, the power requirement for mobile applications needs to be low.

2.6 Multimedia Codecs and Mobile Environments

Although, with packet-switched technology, mobile communications is no longer restricted to voice-only applications, voice will still be the most important service followed by video applications. So, high quality speech and video over IP in mobile environments are two main emerging services. These two applications were used in the simulations throughout this thesis. In this section, the importance of these two scenarios is described. Today, there are various multimedia codecs, which are developed for various reasons for different networks. Three different speech codecs and a video codec, which were used in the simulations, have been described in this section.

2.6.1 Voice over IP (VoIP)

Packet speech communication is one of the crucial applications in 3G networks, which has been the subject of a considerable amount of research. VoIP (voice over IP - that is, voice delivered using the Internet Protocol) is a term used in IP telephony for a set of facilities for managing the delivery of voice information using the Internet Protocol (IP). In general, this means sending voice information in digital form in separate packets rather than in the traditional 2nd generation circuit-switched of the public switched telephone network (PSTN) and GSM mobile system. A major advantage of VoIP and Internet telephony is that it avoids the tolls charged same as ordinary telephone service. In addition to IP, VoIP uses the real-time protocol (RTP) to help ensure that packets get delivered in a timely way. Using public networks, it is currently difficult to guarantee Quality of Service (QoS).

For VoIP application, the choice of speech vocoder has a very important role for transmitting acceptable sound quality. To date, all the speech vocoders are designed based on the circuit-switched network to optimise the speech quality and spectrum efficiency. However, they only consider the bit errors. Whereas, over packet-switched environment, as well as bit error, packet loss may also occur, especially due to the header corruption, which is the main challenge to achieve quality. The main reason of this is, the total amount of required overhead, including RTP, UDP and IPv4, is 40 bytes, which represent the majority part of the packet compared to the speech data that the packet contains. Because of that, it is very likely that the header parts get errors, and cause packet loss.

G.723.1 is the speech codec that is used today in Internet telephony applications, such as netmeeting. G.729b with AMR are the two speech vocoders, resulting good performance over mobile channels [36][37][97]. These speech vocoders have been used in the simulations carried out in this study to measure the header compression performance effect on their quality. In the following sub-sections, G.723.1, G.729b and AMR vocoders are explained in more details.

2.6.1.1 G.723.1

G.723.1 [35] is the ITU-T Recommendation, which was standardised in 1996, for compressing the speech and audio signal component of multimedia services at a very low bit rate. It is a dual rate codec for multimedia communications transmitting at 5.3 and 6.3 kbit/s, which are considerably low. The higher bit-rate has better quality, whereas the lower bit rate gives good quality and provides system designers with additional flexibility. In the design of the codec, the principal application considered was very low bit rate visual telephony as part of the overall H.323 ITU-T multimedia system [34]. Today, it is widely used over Internet applications, and it is integrated in many services, such as netmeeting, see me-see you.

The coder operates on speech frames of 30 ms corresponding to 240 samples at a sampling rate of 8 kHz. For the low bit rate codec, 5.3 kbit/s, an encoded frame requires 158 bits. For the higher rate, 6.3 kbit/s, an encoded frame requires 189 bits. Additionally, 2 control bits are needed for each frame, to indicate whether the high (0) or low (1) rate is used, and to indicate whether the current frame is active speech (0) or non-speech (1). Thus 20 octets are required for low bit rate codec, whereas 24 octets (including 1 unused bit) are required for the high bit rate codec. This means, mapping one-frame-to-one RTP packet, for low rate about 33 % and for high rate 37.5 % of the packet will be the payload, and the rest will be RTP/UDP/IP overhead. Using more than one speech frame in one RTP packet, reduces the average overhead requirement, however every extra speech frame used in a packet will cause extra 30 ms delay, which is the one frame length. In addition in that case, any packet loss will cause two speech frames, 60 ms long, to be lost at a time. Therefore in the simulation in this thesis, one speech frame was transported in one RTP packet. The performance of the G.723.1 codec over GPRS mobile channels is presented in future chapters.

2.6.1.2 G.729b

G.729b [26] is one of the ITU-T standards for coding of speech signals at 8 kbit/s using Algebraic-Code-Excited Linear-Prediction (ACELP). The coder is a reduced complexity version of the full G.729 speech codec. It has been developed for multimedia simultaneous voice and data applications, although the use of the codec is not limited to these applications.

The G.729b coder is designed to operate with a digital signal obtained by first performing telephone bandwidth filtering (G.712) [159] of the analogue input signal, then sampling it at 8000 Hz, followed by conversion to 16-bit linear PCM for the input to the encoder. The output of the decoder should be converted back to an analogue signal by similar means.

The coder operates on speech frames of 10 ms corresponding to 80 samples at a sampling rate of 8000 samples per second. This means, mapping one-frame-to-one RTP packet, only 20 % of the packet will be the payload, and the rest will be RTP/UDP/IP overhead. G.729b is one of the speech codecs that is used along side the header compression schemes simulations. The performance of the G.729b codec, when different header compression algorithm is employed, is presented in future chapters.

2.6.1.3 AMR

Adaptive Multi-Rate (AMR) [38] is a low bandwidth voice encoder/decoder (vocoder). Initially, it is designed especially for use in 2nd generation GSM wireless networks by the European Telecommunication Standards Institute (ETSI). Furthermore, today it has been designated as the default vocoder for 3rd Generation networks (3G) by the 3G Partnership Project (3GPP), which produces technical specifications for 3G networks.

It is based on the Algebraic Code-Excited Linear-Prediction (CELP), same as G.729b speech vocoder, but it is referred to as a Multi-Rate ACELP. The coder is capable of operating at 8 different bit-rates denoted coder modes, from 4.75-to-12.2 kbit/s. It has been designed that the bit rate can be changed every 20 ms, which is also the frame rate of AMR vocoder. It adapts to the network channel conditions. The wireless environment is an inhospitable place for information transmission due to noisy, varying signal level, and high bit error rate (BER) channel conditions, which are caused by multi-path fading and continuous mobility. In these channel conditions, AMR vocoder uses higher bit rate, when the reception between the user and the network is good, as there is less chance of errors or missed frames. When, the channel condition changes or when the user moves to an area with worse reception, AMR vocoder can progressively reduced its bit rate, and extra channel coding is used, so that errors do not cause severe degradation.

AMR, like many other vocoders, uses Cyclic Redundancy Check (CRC) to verify that a block of data is sent correctly. The receiving side compares the CRC bits, to detect if an error is introduced during transmission. If they do not match, the block of data is then known to be errored and can be thrown out. However, within the transmission data there are some portions, that the error on them is insignificant to the human ear. Therefore, AMR vocoder divides every 20 ms speech frame into three groups, which are A, B, and C bits, according to their importance for subjective quality. The A group bits are essential to reproducing accurate sound, and C group bits are the least essential. If an error occurs in the A group bits, then the frame is thrown away. Whereas, errors in B and C group bits do not severely degrade the speech quality, so in these cases, the frame is played out normally. This results in better sound quality, since fewer packets are caused to be discarded. Hence, all these advantage put AMR vocoder in most favourable position against the other speech vocoder for future wireless environment. It is certain that, in most 3G wireless networks around the world, AMR will be one of the most heavily used vocoders. In fact, it is very likely that current 2G networks will begin implementing AMR, especially as a part of the evolution process to 3G.

2.6.2 Video over IP

The delivery of video over mobile networks is far from straightforward than the transmitting of speech, and has been the subject of a considerable amount of research. Video transmission requires much more bandwidth than speech transmission, which is not supported by the second-generation circuit-switched GSM networks. Current circuit-switched networks would only allow bandwidths for video of around 9 kbit/s. To meet these low bandwidth requirements, the video would have to be encoded at very low quality, along with very low frame rates. So, the limited bandwidth is one of the principal barriers to the implementation of video services.

So, an emerging service of future multi-service networks is video communication over packet-switched environments. The recent development in video compression standards such as H.261 [151], H.263 [152], MPEG-1 [160], MPEG-2 [161], and MPEG-4 [154] [155] has made it feasible to transport video over computer communication networks, as well as mobile channels. Video images are represented by a series of frames in which the motion of the scene is reflected in small changes in sequentially displayed frames. Frames are displayed at the terminal at some constant rate (e.g. 30 frame/s) enabling the human eye to integrate the differences within the frame into a moving scene.

In terms of the amount of bandwidth consumed, video transmission is still high on the list even though, today MPEG-4 provides robust and acceptable video transmission at 48 kbit/s. Apart from the high throughput requirements, video applications also put a stringent requirement on packet loss and delay. Also, MPEG-4 supports variable bit rate video encoding.

2.6.2.1 MPEG-4

In this thesis, full error resilient MPEG-4 video codec, which is developed within multimedia group, is used. MPEG-4 Visual is a visual coding standard with many new features: high coding efficiency; high error resilience; multiple, arbitrary shape object-based coding. It covers a wide range of bit-rates from scores of Kbps to several Mbps. It also covers a wide variety of networks, ranging from those guaranteed to be almost error-free to mobile networks with high error rates. Video compression in MPEG-4 is based on exploiting spatial and temporal data redundancies in video frames. The human eye is incapable of resolving high frequency colour changes. Hence, these changes are not transmitted to save on bandwidth. Discrete Cosine Transform (DCT) is used, along with quantisation and Huffman coding to predict a pixel value from all adjacent pixel values. This generates the Intra-frames or I-frames. Motion compensation and prediction process then predicts the value of pixels in a frame from the information in adjacent frames. These will then be coded as Predictive-frame (P-frame) or Bi-directional frame (B-frame). An I-frame contains full picture information and is the least compressed. It can be transmitted periodically or when there is very high motion in the scene or the frame is different from preceding frames and cannot be predicted from the previous frame. P-frames are then predicted from previous I-, or P-frames with only the difference between the prediction and the actual frame being coded. B-frames use past and future I-, and P-frames for motion compensation. The scene can be broken down into several Group of Pictures (GOPs), which consist of the I-, P-, and B-frames. This is shown in Figure 2-12.

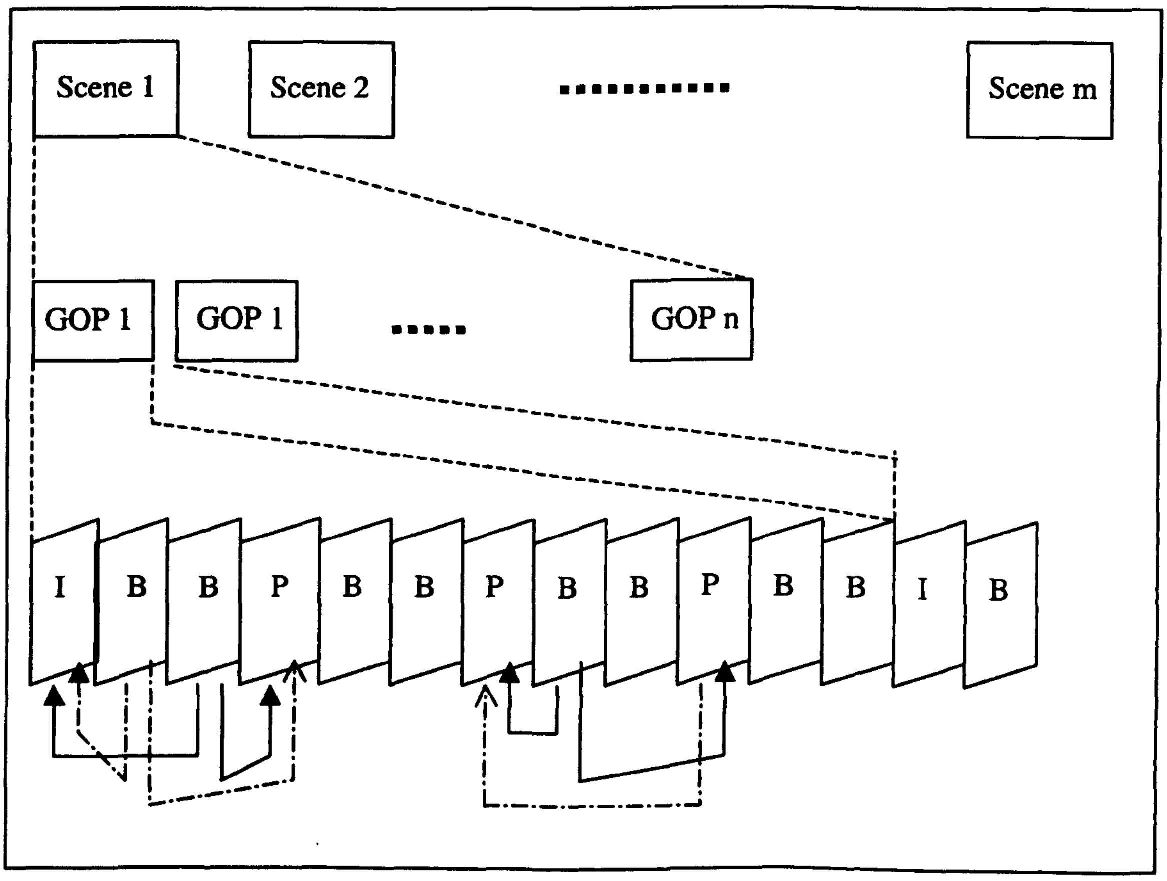


Figure 2-12: MPEG-4 Video coding stream sample

For simplicity, the MPEG-4 video frames are used with one-frame-to-one RTP transmission packet mapping in the simulation that is carried out in this thesis. However, the fragmentation can always be applied without any difficulty.

MPEG-4 is used as a default for a wide variety of networks. One of the main challenging issues for video transmitting over IP across mobile links is how to packetise the frame. Since, the number of bits per frame can be too high, therefore, "a single video packet shall always be mapped on a single RTP packet" may be inappropriate. It is desirable not to apply too much restriction on fragmentation. On the other hand, careless, media unaware fragmentation may cause degradation in error resilience and bandwidth efficiency. The fragmentation rules should be flexible but manage to define the minimum rules for preventing meaningless fragmentation while utilizing the error resilience functionalities of MPEG-4. Hence, the semantics of RTP headers in such cases need to be clearly defined, including the association with MPEG-4 video data elements. In addition, it is beneficial to define the fragmentation rules of RTP packets for MPEG-4 Video streams so as to enhance error resilience by utilizing the error resilience tools provided inside the MPEG-4 Video stream.

From the brief description of MPEG and from Figure 2-12, it can be seen that there is high correlation between the frames (although not all relationship between the frames are shown in the diagram above). These correlations arise because of visual similarities between consecutive images (or parts of images) in a video stream. For a detailed description of MPEG-4 please refer to [154] [155].

2.7 Multimedia Communications Standards

This section is focused on the systems that enable real-time multimedia communication over packet-based networks. The packet-switched technology brings tremendous multimedia applications flexibility and extensibility to the wireless environments. In 3G technology, there are many protocols, which are used for different multimedia applications with different purposes. With packet-switched technology, service control systems are developed. Initially, H.323 system had been developed to enable real-time multimedia transmission, and then SIP was developed as an application layer signalling system, which is used to establish, modify and terminate calls. Both of these systems provide a service control infrastructure to delivery multimedia services.

2.7.1 H.323

H.323 [9] [18] [29] [34] is an ITU standard system that enables real-time, multimedia communication over packet-based networks (i.e. IP networks). It is designed to extend traditionally circuit-based audiovisual and multimedia conferencing services into packet (i.e. IP-based) corporate networks. H.323 is the most widely used protocol for PC-based conferences.

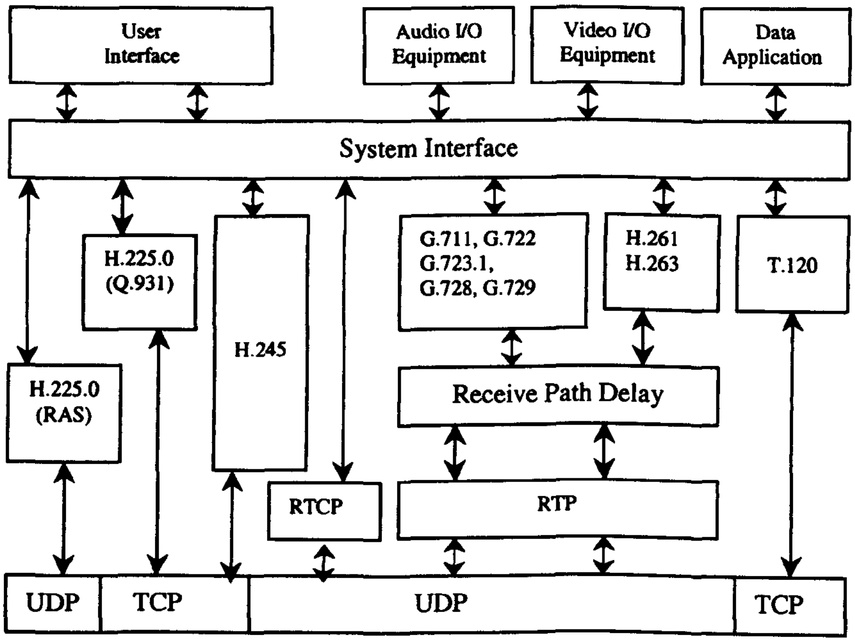


Figure 2-13: H.323 endpoint architecture

The H.323 system aggregates a number of standards; see Figure 2-13, which together allow establishing and controlling point-to-point calls as well as multipoint conferences. Personal computers and other devices-regardless of the hardware, operating system, and software employed- can inter-operate sharing a rich mixture of audio, video, and across all forms of packet-based networks. Seamless interoperation with systems on circuit-switched networks is supported via Gateways. H.323 provides a tightly controlled communications model, with explicit control and media connections set up between participants. Media transmission may occur point-to-point via unicast or take advantage of multicast capabilities of the underlying networks.

The selection of available media, their respective formats, and the transmission topology are dynamically negotiated. In addition to interactive multimedia conferencing, H.323 also has specific provisions for other forms of communication, such as multimedia streaming, distance learning, and IP telephony [9]. As each of these models of communication combines in a different manner, H.323 enables both "join" and "invite" modes in establishing communications. Finally, H.323 defines mechanisms to integrate directory functions, admission control, and call routing that allow implementations (and eventually administrators/users) to define virtually arbitrary usage policies for the H.323 environment.

2.7.2 SIP

The Session Initiation Protocol (SIP) [111] [121] [145] is an application-layer control (signalling) protocol, which is H323 system's alternative, that can establish, modify and terminate multimedia sessions such as Internet telephony calls, with one or more participants, simpler than the H.323. It is one of the IETF standards, which is developed by considering the infrastructure of OSI protocol stack [137] [140] and packet-switched technologies.

Therefore, SIP service technologies can be integrated into Internet technologies that are used widely today. Through the adoption of SIP services, the 3G network will be able to provide rapid and intensive services. There are many Internet applications that require the establishment and management of a session, where a session is considered an exchange of data between groups of users. The users may be move between endpoints, they may be addressable by multiple names and also they may communicate in several different media, sometimes simultaneously. There are a number of protocols that carry various forms of real-time multimedia applications data, such as voice, video, or text messages. SIP works in concert with these protocols by enabling Internet endpoints (called "user agents") to discover one another and to agree on a characterisation of a session they would like to share. For locating prospective session participants, SIP relies on an infrastructure of network hosts (called "proxy servers") to which user agents can send registrations, invitations to sessions and other requests. SIP is a general-purpose tool for creating/terminating sessions that works independently of underlying transport protocols and without dependency on the type of session that is being established.

SIP can also invite participants to already existing sessions. A SIP entity issuing an invitation for an already existing session does not necessarily have to be a member of the session to which it is inviting. Media can be added (and removed from) an existing session. SIP transparently supports name mapping and redirection services, which supports personal mobility.

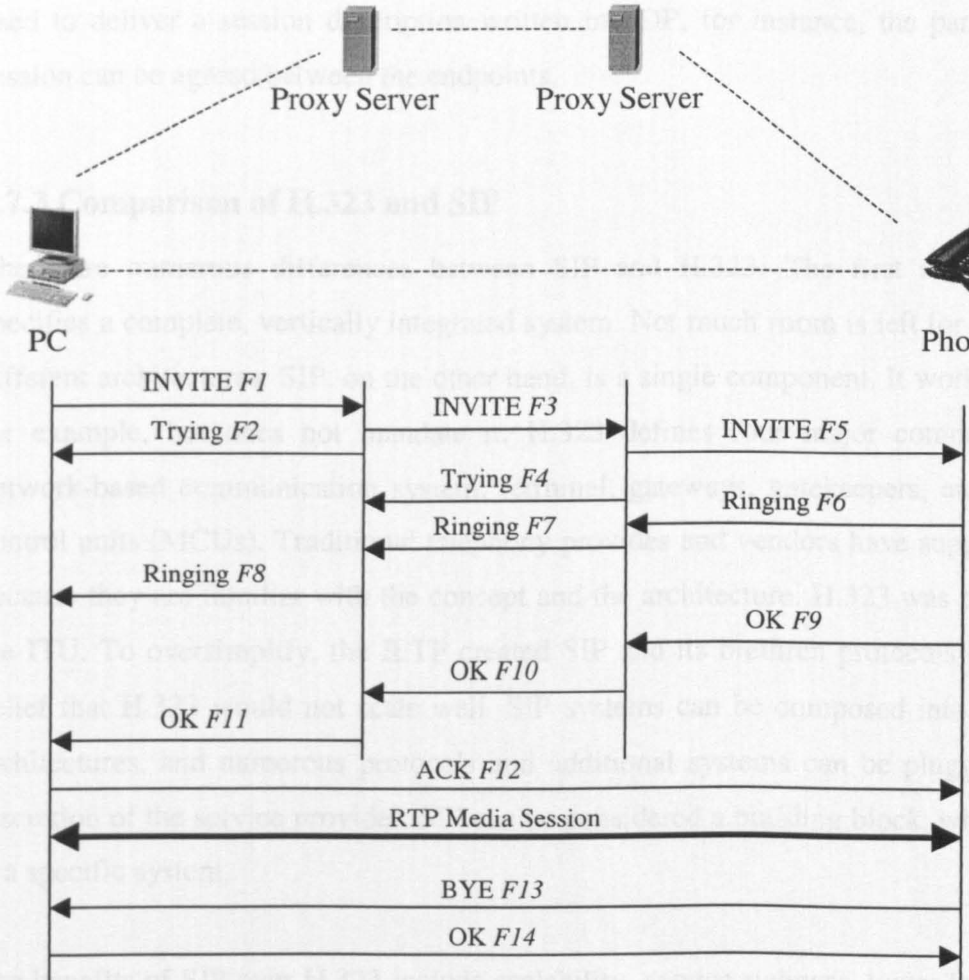


Figure 2-14: SIP session set up example procedure

So, SIP is not a vertically integrated communications system. SIP is rather a component of the overall multimedia data and control architecture which incorporates protocols such as RSVP [151] for reserving network resources, the real-time protocol (RTP) [12] for transporting real-time data and providing QoS feedback, the real-time streaming protocol (RTSP) [148] for controlling delivery of streaming media, and the session description protocol (SDP) [150] for describing multimedia sessions. Therefore, SIP should be used in conjunction with other protocols in order to provide complete services to the users.

However, the basic functionality and operation of SIP does not depend on any of these protocols.

Figure 2-14 presents the SIP session set-up procedure. SIP does not provide services, it provides primitives that can be used to implement different services. For example, SIP can locate a user and deliver an opaque object to his current location. If this primitive is used to deliver a session description written in SDP, for instance, the parameters of a session can be agreed between the endpoints.

2.7.3 Comparison of H.323 and SIP

There are numerous differences between SIP and H.323. The first is scope; H.323 specifies a complete, vertically integrated system. Not much room is left for flexibility or different architectures. SIP, on the other hand, is a single component. It works with RTP, for example, but does not mandate it. H.323 defines four major components for a network-based communication system: terminal, gateways, gatekeepers, and multipoint control units (MCUs). Traditional telephony provides and vendors have supported H.323 because they are familiar with the concept and the architecture. H.323 was developed by the ITU. To oversimplify, the IETF created SIP and its brethren protocols because of a belief that H.323 would not scale well. SIP systems can be composed into a variety of architectures, and numerous protocols and additional systems can be plugged in at the discretion of the service provider. SIP can be considered a building block, whereas H.323 is a specific system.

The benefits of SIP over H.323 include scalability, service richness, lower latency, faster speed, and ability to distribute for carrier-grade reliability. The flip side of this determinism is that H.323 does numerous things that H.323 does numerous things that SIP, purposefully, does not address. H.323 was originally conceived for use on a single LAN [], a LAN protocol. Therefore, numerous enhancements were added to address usage as a wide-area protocol. SIP, in contrast, was designed from day one as a wide-area protocol. SIP's support for fast, stateless proxies in the core, and call stateful proxies in the periphery, adds significant scalability here.

The main advantage of SIP is its full integration with other Internet protocols and functions; SIP is, more or less, equivalent to the Q.931 [157] and H.225 [158] components of H.323. These protocols are responsible for full setup and call signalling. Consequently, both SIP and H.323 can be used as signalling protocols in IP networks.

H.323 and SIP protocols both provide mechanisms for call establishment and tear-down, call control and supplementary services, and capability exchange. Currently H.323 is the most widely used protocol for PC-based conferences, while carrier networking using so-called soft switches and IP telephones seems to be built based on SIP. In order to achieve universal connectivity, interworking between the two protocols is desirable. Interworking between the protocols is made simpler since both operate over IP (Internet Protocol) and use RTP for transferring real-time audio/video data, reducing the task of translation between the signalling protocols and session description.

2.8 Conclusion

This Chapter has given an overview of the current and future communication technologies with enabling multimedia services over mobile access links. It has been made clear that packet-switched technology will be the core technology for these services. In particular, the role of IP protocol stacks, their flexibility advantages, and especially their major impact on mobile communications has been discussed. The main concerns with IP protocol stack usage over bandwidth-limited channels, has also been highlighted.

This chapter has discussed different protocol stacks, which are used for various purposes. Moreover two major applications, Speech and Video are summarised. In addition the important issues that need to be considered for packet-switched technology in mobile channels are mentioned. The remainder of this thesis will introduce different header compression schemes, and examine their performance across the mobile networks. Additionally, the effects of error resilience have been examined. GPRS was chosen as the access network for use in the experiments described in this thesis, as it provides a suitable migration pathway from GSM networks to UMTS.

Chapter 3

3 Enhanced Compressing RTP/UDP/IP (CRTP)

3.1 Introduction

Third-generation wireless systems are designed to offer a multitude of services, providing considerable flexibility, structured QoS handling and cost-effective access, while ensuring coverage with high radio spectrum efficiency. One of the most critical aspects of transmitting real-time multimedia information by using RTP over packet-switched mobile networks is the significantly large RTP/UDP/IP overhead [14] [23] [37], as compared with the limited throughput associated with mobile channels. In addition, the throughput of mobile channels and their error characteristics are time-varying. Therefore, a fundamental challenge is to reduce the size of the RTP/UDP/IP headers, while maintaining the transparency of all header fields. This chapter examines the characteristics of the header fields in detail, the way they change during multimedia transmission, and the advantages of the header compression. To examine the characteristics of the header fields, a header 'capture and analyser' program is implemented in a Microsoft Windows environment.

In this chapter, the Compressing RTP/UDP/IP (CRTP) header compression technique [1] [81] is introduced. An implementation of the compression scheme was designed by the author. An improvement to the CRTP header compression algorithm, which makes use of a reference-based compression, is introduced. Also, an error resilience algorithm, called slow update scheme, is introduced by the author. This scheme increases the robustness, especially where the Round Trip Time (RTT) is high. The experiments were carried out over simulated GPRS mobile access channels. Different speech codecs and the MPEG-4 video codec are tested to provide performance results for standard CRTP and enhanced-CRTP, which are also presented. Experiments are carried out to assess the performance improvement provided by the enhanced CRTP scheme in terms of the perceptual impact upon the decoded speech and video quality. Finally, the comparisons between standard IETF CRTP and enhanced CRTP are provided.

3.2 Utilizing Wireless Channels Effectively

The scenario that has been considered in the experiments described in this chapter, involves two mobile terminals, communicating with each other using packet-switched IP based networks. The terminals are connected to a base station over cellular links, and the base stations are connected to each other through a wired (or possibly wireless) network. So at least one end requires an air interface between base station and mobile terminal (see Figure 3-1).

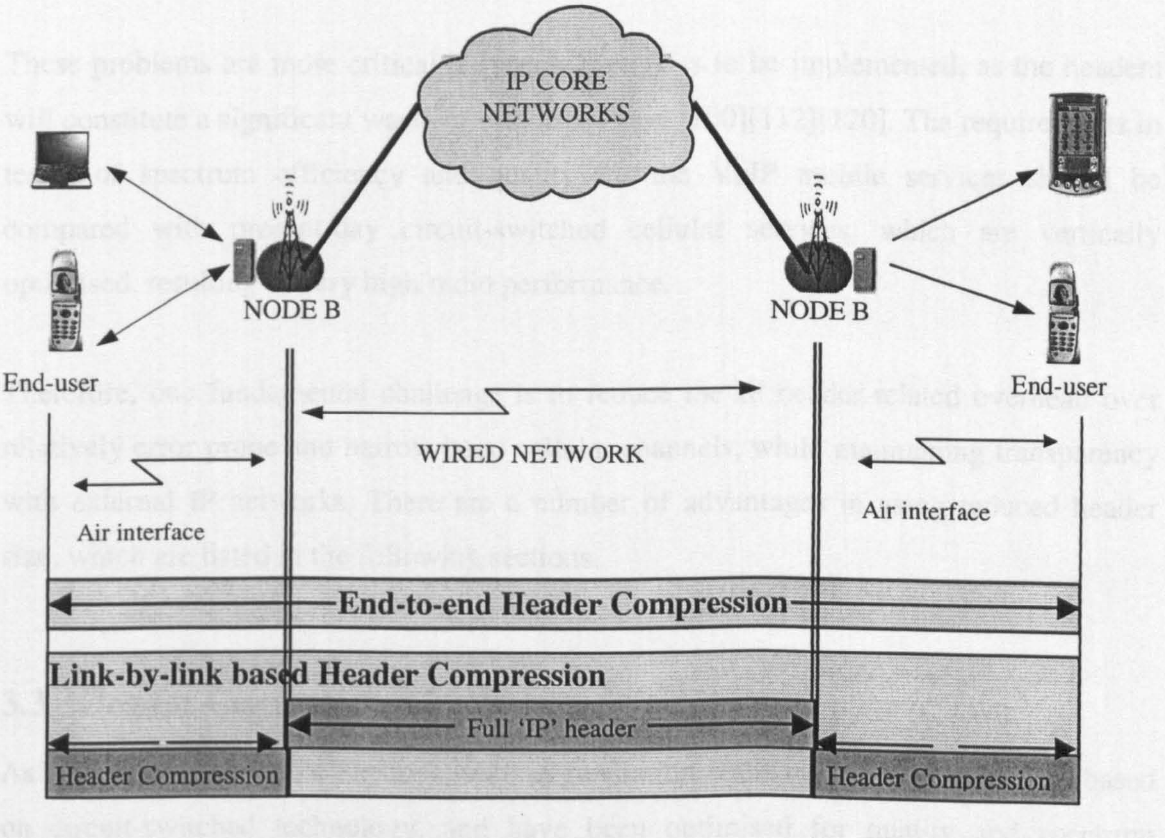


Figure 3-1: Scenario for 'IP' telephony over mobile environment

In IP networks, the combined RTP/UDP/IP headers have a length of at least 40 bytes. This includes the IP header (20 octets), the UDP header (8 octets) and the RTP header (12 octets). If IPv6 is used, the total is increased to 60 bytes. When operating over low throughpüt links, or when transmitting speech or audio streams, which have been compressed to low data rates, this represents a considerable proportion of the total throughput, thereby decreasing transmission efficiency.

The second important characteristic is the lossy nature of cellular links. Wireless networks usually have much higher residual bit error rates (BER) than wired networks. The BER in a wired link is typically less than 1×10^{-12}, whereas in mobile speech channels, bit error rates (BER) can be as high as 1×10^{-2} [13][40]. This high error rate is caused by a shadow fading, multipath fading, and continuous mobility, as well as by inter-user interference from other terminals. In addition, because the overheads are significantly large, there is a high probability that an error occurs on them. Any error on the RTP/UDP/IPv4 headers will cause the packet to be lost, even if there is no error on the payload. This increases the packet loss rate (PLR) and frame error rate (FER) over the network, which affects the quality of service.

These problems are more critical if speech over IP is to be implemented, as the headers will constitute a significant waste of radio spectrum [100][112][120]. The requirements in terms of spectrum efficiency and quality for the VoIP mobile services should be compared with present-day circuit-switched cellular services, which are vertically optimised, resulting in very high radio performance.

Therefore, one fundamental challenge is to reduce the IP header-related overhead over relatively error prone and narrow band cellular channels, while maintaining transparency with external IP networks. There are a number of advantages in using reduced header size, which are listed in the following sections.

3.3 Header Compression Techniques

As discussed in previous Sections, second generation cellular access networks are based on circuit-switched technology, and have been optimised for quality and spectrum efficiency. This results in highly efficient voice services but very low service flexibility.

Therefore, high service flexibility is a priority for 3G and future communication networks. Such flexibility can be achieved by using end-to-end packet-based IP over wireless links [22] [23]. Moreover, Release 99 UMTS networks will provide voice & video telephony over circuit-switched bearers [9] [25] [26 [135].

The size of the payload depends on the speech/video encoding used and the packet rate. It can be as low as 15-20 Octets for speech applications. Clearly, this means that the packet headers introduce a large amount of overhead, which reduces spectrum efficiency. Also, because each of these protocols adds an extra degree of error susceptibility, [see Figure 3-2] the packet error rate significantly increases, particularly in the presence of varying channel conditions.

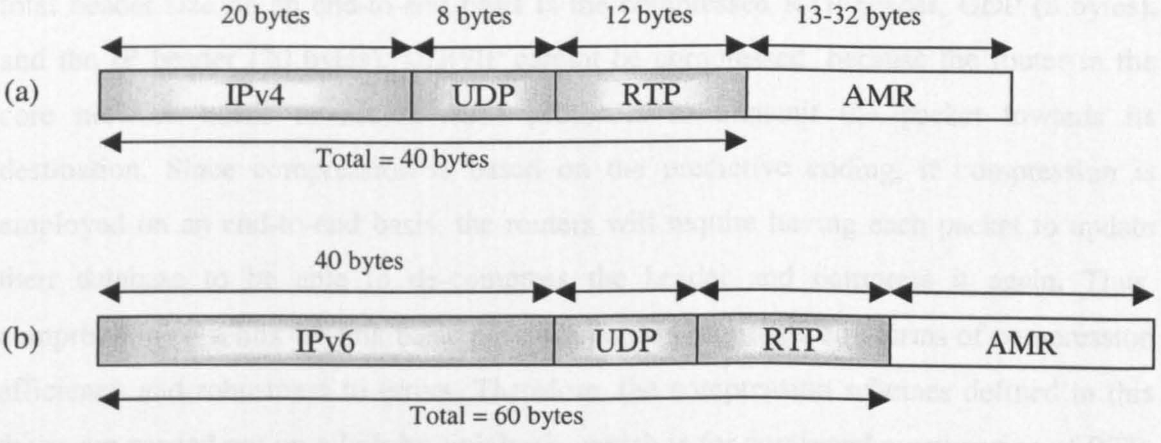


Figure 3-2: AMR speech frame with full protocol stack, IPv4/UDP/RTP (a) and IPv6/UDP/RTP (b)

Consider the case where a voice codec operates at a frame rate (e.g. EFR or AMR), of 50 frames/second (20 ms inter-frame spacing). These frames may be encapsulated into discrete RTP packets using one-to-one frame to packet mapping. In this case, the RTP/UDP/IPv4 headers (40 octets) and RTP/UDP/IPv6 headers (60 octets) alone consume 16kbit/s and 24 kbit/s respectively. This is not efficient for wireless environments, where bandwidth is a limited and expensive resource. When transmitting speech or audio streams, which have been compressed to low data rates, this represents a considerable proportion of the total throughput, thereby decreasing transmission efficiency, as shown in Figure 3-2. So, one fundamental challenge is to reduce the IP header related overhead over the relatively error prone and narrow band cellular channels, while maintaining the transparency of all header fields. Therefore, for real-time applications header compression over wireless links is essential, not optional.

Two methods of providing header compression have been considered. The first is employed on an end-to-end basis (see Figure 3-1), where compression is applied to the RTP header alone. This is because the routers need to know IP and UDP header information to decide where to forward the packet, so as to avoid decompression and compression at each router within the core network, the compression can only be applied to the RTP header alone. The second is on a link-by-link basis (see Figure 3-1), where compression is applied to the combined RTP, UDP and IP headers. Compression on a link-by-link basis provides a number of advantages against compression on an end-to-end basis. End-to-end compression operates on the 12 bytes of the RTP header alone. The total header size on an end-to-end basis is the compressed RTP header, UDP (8 bytes), and the IP header (20 bytes). UDP/IP cannot be compressed, because the router in the core network needs to access these protocols to transmit the packet towards its destination. Since compression is based on the predictive coding, if compression is employed on an end-to-end basis, the routers will require having each packet to update their database to be able to de-compress the header and compress it again. Thus, compression on a link-by-link basis provides better performance in terms of compression efficiency and robustness to errors. Therefore, the compression schemes defined in this thesis are carried out on a link-by-link basis, which is for combined compression of RTP, UDP, and IP headers.

A number of combined RTP/UDP/IP compression algorithms have been proposed, including CRTP [1] and ROCCO [6]. The following sections focus on the CRTP header compression algorithm and techniques for improving the robustness of the algorithm.

3.4 Advantage of Header Compression

When the overhead of RTP/UDP/IP protocols is analysed, a high degree of redundancy between fields in the headers of consecutive packets belonging to the same packet stream has been observed. This observation is the basis for header compression. Using header compression provides many advantages for transmission over communications networks. These can be summarised as follows:

* *Improve interactive response time*

For very low-speed links, echoing of characters may take longer than 100-200 ms because of the time required to transmit large headers. 100-200 ms is the maximum time people can tolerate without feeling that the system is slow. For satellite this time can go up to 500-600 ms from end-to-end.

* *Allow use of small packets for bulk data with good line efficiency*

This is important when interactive (such as Telnet) and bulk traffic (for example multimedia) is mixed because the bulk data should be carried in small packets to decrease the waiting time when a packet with interactive data is caught behind a bulk data packet.

* *Allow use of small packets for delay sensitive low data-rate traffic*

For such applications, for example voice, the time to fill a packet with data is significant if packets are large. To get low end-to-end delay small packets are preferred. As described in section 3.3 above, without header compression, the figures show that, to provide IP speech telephony, will further increase the bandwidth consumed by headers by 16-24 kbit/s. This should be compared with the bandwidth required for the actual sound samples (e.g. 13 kbit/s with GSM encoding). CRTP header compression can reduce the bandwidth needed for headers significantly, in the voice example the header bit rate can be reduced to about 0.8 kbit/s with the Ideal case scheme that is explained in the second paragraph of section 3.6.4. This enables higher quality voice transmission over low throughput channels. However, this scheme is not practicable.

* *Reduce packet loss rate over lossy links.*

Because fewer bits are sent per packet, the packet loss rate due to the header corruption will be lower for a given bit-error rate. This results in a higher throughput for data, as the successful packet-sending window can open up more between losses, and in fewer lost packets from header corruption

3.5 Experimental Setup

The first experiment is setup to carry out real audio and video RTP packet transmission over network. Basic IP connections are implemented. UDP is usually exploited through the use of Microsoft Window programming (WinSock). The simplicity of implementing connections using sockets comes at the expense of limited access to IP and UDP header parameters. Most of the values are assigned automatically by the libraries. The RTP implementation is written in C++, and exploits the UDP sockets libraries. The availability of the source code means that much greater control is possible over the RTP part of the connection than any other. Once the connection is established between two PCs, speech and video data is packetised using RTP/UDP/IPv4 protocols and is transmitted from one PC to another PC via the internal network of CCSR (department network) (see Figure 3-3). Having said that packets are sent through the network, all packets within the same hub are accessible by the host (PC), the hosts (PCs) need to check the address on the RTP packet. In this way the packets will be accepted by the hosts with matching address. In order to capture each packet for analysis, a network packets capture and analyser program was implemented, as shown in Figure 3-4. Packet Capture collects packets from the network segment that the Network Interface Card (NIC) is connected to. It can start to capture a new packet whilst simultaneously analysing a previous packet.

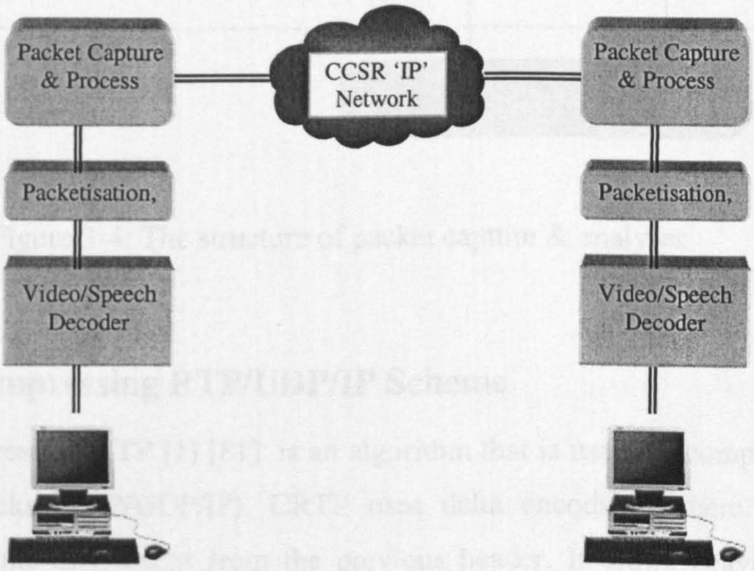


Figure 3-3: Implemented Packet Capture Experiment Scenario

This program can however only view the captured packet and collect traffic from the segment where the analyser is located. It is also possible to capture and analyse traffic from another segment (local multi-segment LAN or remote WAN), by means of a distributed or multi-segment analyser. Distributed analysers offer similar functionality to a standard (non-distributed) analyser, displaying multiple diagnostic windows, each representing a segment on your LAN - all from a single management station. Typically, distributed analysers consist of a software-based management station and either software or hardware based probes, which allow an administrator to "view" any segment that hosts a probe.

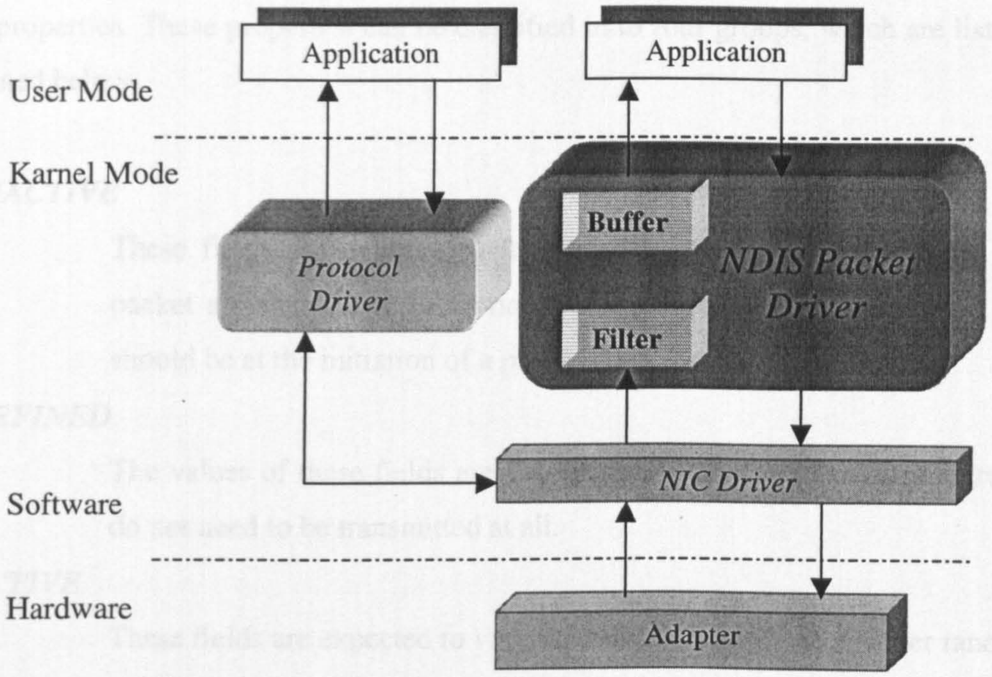


Figure 3-4: The structure of packet capture & analyser

3.6 CRTP-Compressing RTP/UDP/IP Scheme

The CRTP, Compression-RTP [1] [81], is an algorithm that is used for compressing real-time protocol stacks (RTP/UDP/IP). CRTP uses delta encoding, where compressed headers represent the differences from the previous header. It draws heavily from the TCP/IP header compression algorithm [7]. Originally, the CRTP compression scheme was designed to address the specific problem of sending audio and video over dialup modem links, whose throughputs range from 14.4 to 56 kb/s. These links generally require full-duplex communication. It is shown that the CRTP header compression scheme reduces the RTP/UDP/IP headers to a minimum of two bytes when the UDP checksum is not enabled, and if enabled, the minimum compressed header is 4 octets. The details are given in the following sections.

The result of the packet capture observations from section 3.5 and listed in section 3.6, have shown that header compression is possible due to the fact that there is much redundancy between field values within packets. Most of the redundancy is present between consecutive packets. For header compression, it is very important to make use of these properties. These properties can be classified in to four groups, which are listed and explained below:

- *INACTIVE*

  These fields are expected to be constant throughout the lifetime of the packet stream. These fields need to be transmitted at least once, which should be at the initiation of a packet stream.
- *DEFINED*

  The values of these fields are expected to be well-known. Therefore, they do not need to be transmitted at all.
- *ACTIVE*

  These fields are expected to vary in an unpredictable way, either randomly, or within a limited value set or range.
- *INFERRED*

  These fields contain a value that can be inferred from other values.

The names, "INACTIVE and ACTIVE" are given by the author, and the others are taken from [1] [6]. The results of the packet capture observations showed most of the fields remain unchanged (DEFINED and INACTIVE), and several fields change in every packet (ACTIVE). The ACTIVE fields can be split into three categories. The first category represents the fields that change completely at random, and the second represents the fields that have very little change from packet to packet. Finally, the third category represents the fields that have a constant change from packet to packet, implying that the second-order difference is zero. Also, there are some fields contain a value that can be inferred from other values. Such fields can be classified as INFERRED.

As shown above, there is a lot of redundancy between header fields, both within the same packet header and also between consecutive packets belonging to the same packet stream.

- 51.75% of these overheads are total of DEFINED and INACTIVE fields, 10% are INFERRED and only 38.25% are ACTIVE.

The above figures represent account all the fields of RTP/UDP/IP, which are listed in the tables in section 3.6.1. By sending the information held in the DEFINED and INACTIVE fields at the initiation of a packet stream, and utilising dependencies and predictabilities for other fields, the header size can be significantly reduced for most packets.

In the CRTP scheme, a significant compression gain is achieved using the fact that the differences in several fields between successive packets are constant, thereby representing a second-order difference of zero. This means that the transmitter needs only to send an initial uncompressed header and first-order differences, followed by indications of which fields in subsequent packets have zero second-order differences. The de-compressor must therefore maintain a context consisting of the most recent uncompressed header received, which it combines with each received compressed RTP/UDP/IP header to reconstruct the original headers.

CRTP maintains a session context, which is essentially the uncompressed version of the most recent full, uncompressed header sent over the link, at both compressor and de-compressor. The session context is defined by the combination of the IP source and destination addresses, the UDP source and destination ports, and the RTP SSRC field. Compression and de-compression are performed relative to the context. The session contexts are stored in tables at both the compressor and de-compressor ends. The compressed packet carries the session context identifier or CID, which is used to indicate the session context in which the packet should be interpreted. The CID is a small integer number. The session context identifier, CID, is either an 8-bit or a 16-bit integer, depending upon the number of contexts. The de-compressor can use the CID to index its table of stored session contexts directly. When compressed headers carry differences between the previous header, and the current one, each compressed header will update the context of the de-compressor. When a packet is lost between the compressor and corresponding de-compressor, the context of the de-compressor will lose synchronisation with the source context, since it is not updated correctly. Both uncompressed and compressed packets carry this CID and a 4-bit sequence number used to detect packet loss between the compressor and de-compressor.

The general principle of the CRTP header compression is to restrict the number of times a packet with a full header needs to be sent; subsequent compressed headers refer to the previous full header and may contain incremental changes from that header.

The shared information in each context consists of the following items:

- The full IP, UDP and RTP headers, possibly including a CSRC list, for the last packet sent by the compressor or reconstructed by the de-compressor.
- The first-order difference for the IPv4 ID field, initialized to 1 whenever an uncompressed IP header for this context is received, and updated each time a delta IPv4 ID field is received in a compressed packet.
- The first-order difference for the RTP timestamp field, initialised to 0 whenever an uncompressed packet for this context is received, and updated each time a delta RTP timestamp field is received in a compressed packet.
- The last value of the 4-bit sequence number, which is used to detect packet loss between the compressor and de-compressor.
- A context-specific delta-encoding table that may optionally be negotiated for each context

In the event where the compressor fails to compress the protocol stacks after a number of attempts due to an unexpected change within a field or fields, the compressor maintains a "negative cache" of packet streams where compression has failed. As real-time applications are inherently very delay-sensitive, the "negative cache" is used to prevent further failed compression attempts. The list is maintained based on compression attempt time rather than a specific number of attempts. Failure of compression usually means that the compressor is faced with unexpected changes in the protocol stacks (mainly in the RTP header). For instance, the compressor and de-compressor design their context table, which keeps the necessary information for compression and de-compression, by taking into account the fact that some fields in the RTP header remain constant most of the time, such as the payload type field, but keep changing at certain instances. The negative cache is indexed by the source and destination IP address and UDP port pairs, but not the RTP SSRC field since the latter may change. The CRTP header compression scheme can compress IP and UDP protocols, even if RTP headers cannot be compressed.

3.6.1 Classification of Real-time Protocol stacks

Using the experimental set-up described in section 3.5, the real-time protocol stacks' fields have been investigated individually to observe the correlation between fields in consecutive packets during the real-time voice and video communications. The results are presented in this section.

3.6.1.1 IPv4

Table 3-1 shows how the IPv4 protocol stack fields are classified. The link layer protocols used by mobile access networks provide services that the CRTP compression scheme may exploit, such as length information and error detection. For example, the *total length* in the IPv4 header can be inferred by using the link layer length information. In addition, because the link layer often provides option for error detection [1], the header *checksum* can be omitted. This leaves the *packet ID*, which does not need to be transmitted if there is no IP fragmentation. To maintain lossless compression, changes in the packet ID are transmitted. Normally, it increases by one or a small amount between consecutive packets.

Field	Size (bits)	Class
Version	4	Inactive
Header Length	4	Defined
Type of Service	8	Active
Packet Length	16	Inferred
Identification	16	Active
Flags	3	Defined
Fragment Offset	13	Defined
Time to Live	8	Active
Protocol	8	Inactive
Header Checksum	16	Active
Source Address	32	Inactive
Destination Address	32	Inactive

Table 3-1: Classification of IPv4 protocol stack

3.6.1.2 UDP

Table 3-2 shows the field classifications of UDP protocol stack. In the UDP header, the main field for concern is the *checksum*, which is not predictable and can considered as a random value, even though it is not strictly random from a mathematical point of view. It must be transmitted intact in order to ensure lossless compression, if it is not set to zero. It is very important for maintaining end-to-end error detection for applications, even though retransmission is not practical for low-delay real-time services. The other field that changes is the *length field*, which is redundant because of the IP total header length field and the fact that the length is also indicated by the link layer.

Field	Size (bits)	Class
Source Port	16	Inactive
Destination Port	16	Inactive
Length	16	Inferred
Checksum	16	Active

Table 3-2: Classification of UDP protocol stack

3.6.1.3 RTP

The Table 3-3 assumes that there is no packet lost and no mis-ordered packets between compressor and de-compressor. The *sequence number* and *timestamp* are major fields in RTP headers. These two fields change from packet to packet. The change in the sequence number is straightforward, increasing by one for each packet. But the timestamp is not as simple. The increment of the timestamp from packet to packet is based on the application and the state of the communication. For speech packets, the timestamp increases by a multiple of the sample period.

Field	Size (bits)	Class
Version	2	Inactive
Padding	1	Inactive
Extension	1	Inactive
CCSR Counter	4	Active
Marker	1	Active
Payload Type	7	Active
Sequence Number	16	Active
Timestamp	32	Active
SSRC	32	Inactive
CSRC	0 (-48)	Active

Table 3-3: Classification of RTP protocol stack

For video, the situation is different. Here, the timestamp only change in the first packet of each frame, assuming that more than one packet is required to transmit a frame. If one-video-frame-to-one-packet mapping is used, then the change from one packet to next packet is constant. In each of these cases the second-order difference of the sequence number and timestamp fields is zero, which means that the field information can be constructed by adding the first-order differences to the previous header's fields. The first-order differences for these fields are stored in the session context along with the previous uncompressed header. When the second-order difference is not zero, the magnitude of the change is transmitted, rather than the absolute value.

The ***Marker bit*** is used to indicate the first packet of the talkspurt and the last packet of a video frame. Finally, the ***CSRC*** list and ***CSRC counter*** can only change if the packet flows through an RTP mixer. However, the CSRC list typically remains constant during a talk spurt or longer, so it need be sent only when it changes. Finally, the ***SSRC Identifier*** field is part of what identifies the particular context, so it is constant for a given context.

3.6.2 Header Formats

In order to communicate using packets whilst using different uncompressed and compressed header formats, the CRTP protocol depends upon the underlying link layer being able to provide an indication of three packet formats in addition to the normal IP packet formats.

Three packet formats are defined in standard:

- *FULL_HEADER*
- *COMPRESSED_UDP*
- *COMPRESSED_RTP*

The proposed new packet format:

- *REFERENCE_HEADER*[‡]

[‡] The reference_header format is introduced by the author

3.6.2.1 Full Header

The format of the FULL_HEADER packet is the same as that of the original packet (i.e, uncompressed). In the scenarios investigated, this is usually an IP header, followed by a UDP header and an RTP header and datagram. The only difference from the original packet is that it carries the CID and the 4-bit sequence number. In order to avoid expanding the size of the header, these values are inserted into the length fields in the IP and UDP headers. The actual packet length can be inferred from the length provided by the appropriate fields of the link layer header. The first length field is the total length field of the IPv4 header, and the second is the length field of the UDP header. When a FULL_HEADER packet is received, the complete set of headers is stored in the context selected by the context ID. The 4-bit sequence number is also stored in the context, thereby re-synchronizing the de-compressor to the compressor. The compressed REFERENCE_HEADER indicator is also stored in this header. The reference_header is a new-presented packet format, whose function is explained in section 3.6.2.4. Six bits are been allocated for its indicator. Once the de-compressor receives this indicator, it knows when the compressed REFERENCE_HEADER will be included.

For 8-bit context ID:

0	1	Indicator	CID	First length field

0	Seq	Second length field

For 16-bit context ID:

1	1	Indicator	0	Seq	First length field

CID	Second length field

Figure 3-5 The format of length fields in the Full-header

3.6.2.2 Compressed RTP Packet

A compressed RTP packet header contains only 2 octets when the second-order difference of the RTP header from packet to packet is zero. The de-compressor can reconstruct a packet simply by adding the stored first-order differences, which is known as DELTA value, to the stored uncompressed header, which represents the previous packet. All that needs to be communicated is the session context identifier and a small sequence number to maintain synchronization and detect packet loss between the compressor and de-compressor. Each time the first-order difference changes, it is transmitted and stored in the context at both the compressor and the de-compressor ends.

The compressed RTP packet format is:

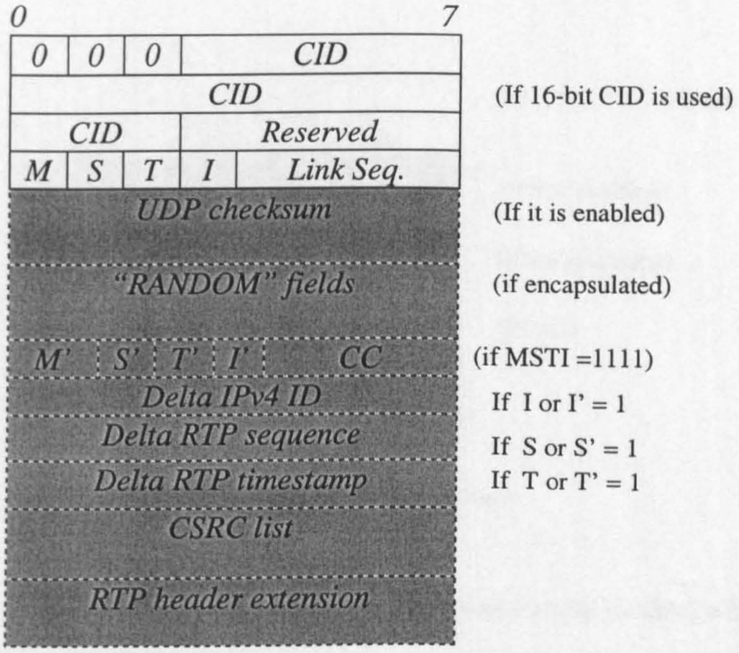


Figure 3-6: Compressed RTP packet format

A bit mask is used to indicate the fields that have changed by a value other than the expected difference. This bit mask contains four bits, and these bits represent the RTP marker bit, the RTP sequence number, the RTP timestamp, and the IPv4 packet ID (MSTI). These fields' delta values, which represent their changes, are stored at both ends, at the compressor and at the de-compressor. If any of these fields' changes are different from their corresponding stored delta value, their representative field within the bit map set to "1". Then the compression format will insert delta fields to represent the change of the following fields.

M = RTP marker bit

S = RTP sequence number

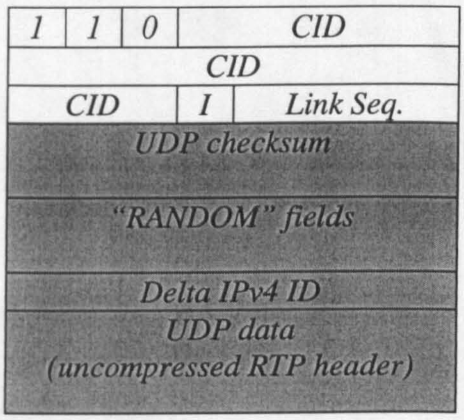
T = RTP timestamp

I = IPv4 packet ID

Figure 3-6 shows the compressed IP/UDP/RTP header with dotted lines indicating fields that are conditionally present. The first three bits represent the packet format, which they named as "Format bit-mask". "All zeros" format bit-mask represents the compressed_RTP packet.

3.6.2.3 Compressed UDP Packet

The format is;



(If 16-bit CID is used)

(If it is enabled)

(If encapsulated)

(if I =1)

Figure 3-7: Compressed UDP packet format

Figure 3-7 shows the compressed UDP packet format. This packet format is used when an RTP header cannot be compressed. This can happen whenever there is a change in any of the fields of the RTP header that are expected to be constant (such as the payload type field). If the IP and UDP headers do not also require updating, the RTP header is carried in a COMPRESSED_UDP packet rather than a FULL_HEADER packet. The COMPRESSED_UDP packet has the same format as the COMPRESSED_RTP packet except that the M, S and T bits are always 0 and the corresponding delta fields are never included, since RTP protocol is sent as-is, without compression.

3.6.2.4 Compressed REFERENCE_HEADER Packet

The Compressed REFERENCE_HEADER packet is a new header format that is proposed by the author. It is used to increase the robustness of transmission. The standard CRTP compression algorithm operates over consecutive packets, which reduces the robustness of the compression over links with long round-trip delays. When, the round-trip-time is larger than the inter (compressed) packet spacing, if one packet's compressed header is corrupted and the de-compressor cannot use it to reconstruct the original header, it will discard the corrupted packet, together with all the outdated packets that arrive before an uncompressed error-free header arrive, see Figure 3-8. The result is that some received packets are invalidated unnecessarily, causing extra bandwidth to be consumed.

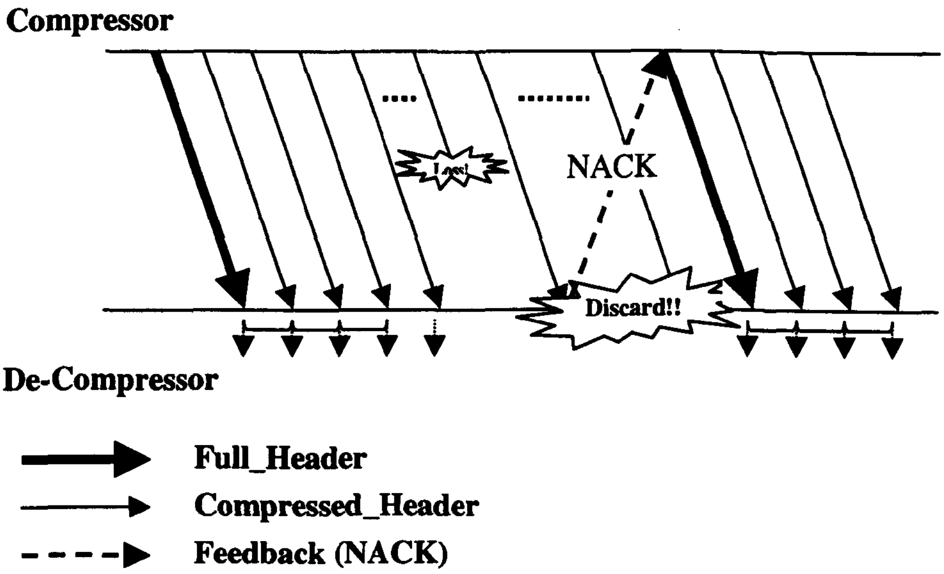


Figure 3-8 Transmission Sequence of CRTP with Reference_Header

The REFERENCE_HEADER packet is used as a reference for the compression. This is shown in Figure 3-9; header compression is not based on the last transmitted consecutive packet anymore. When it is used, the compressed_RTP and the compressed_UDP packets are compressed based on the REFERENCE_HEADER packet's header fields rather than the previous header. In this case, only the corrupted packets that cannot be used to reconstruct the original header will be discarded. Packets following corrupted packets will not be affected, so the error will not propagate. However, this costs an extra overhead.

Another problem with using classic CRTP header compression in mobile networks is that sometimes the physical layer cannot detect the error on the compressed header. The corrupted header is then passed to the PDCP layer[§] in UMTS and SNDCP layer in GPRS, as if it is not corrupted. Therefore, the PDCP layer reconstructs the header incorrectly, and this reconstructed header becomes the reference for the next packet. Since the header is not being updated correctly, all subsequent headers will be incorrectly reconstructed.

[§] The PDCP is layer, where header compressions take place. It is placed just below network layer

This causes a reduction in the application's performance. By using the REFERENCE-HEADER packet, error propagation due to the context damage is minimised.

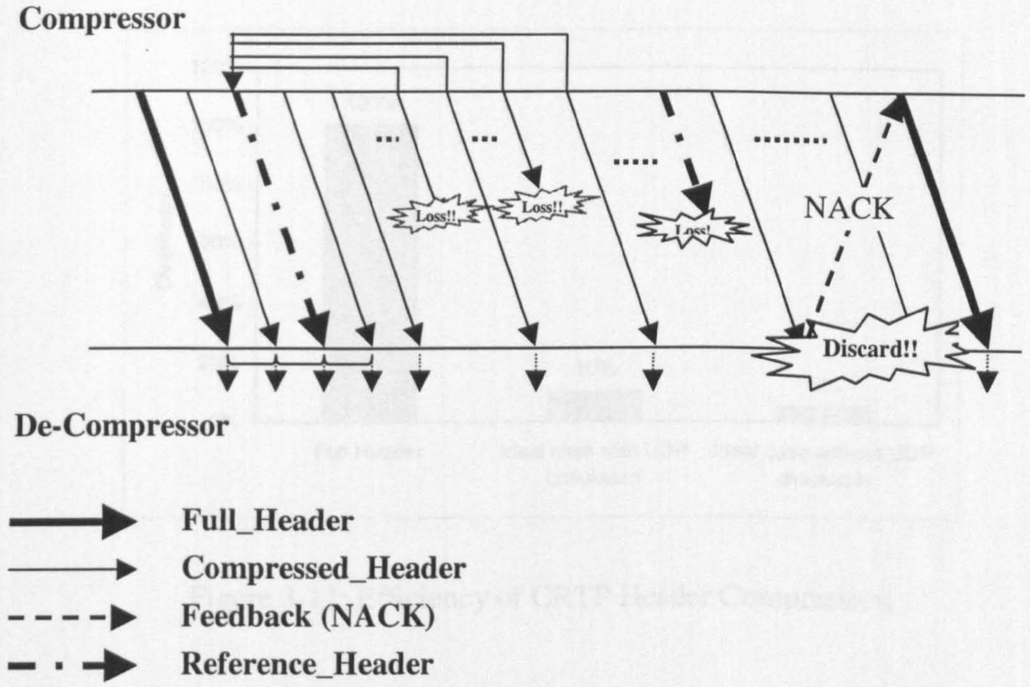


Figure 3-9: Transmission Sequence of CRTP with Reference_Header

The compressed REFERENCE_HEADERs carry the *active* fields' original values, which are shown in Figure 3-10. This packet format is identified by "0 1 1" format bit-mask. The REFERENCE_HEADER packet format is defined as follows;

<table>
<tr><td>0</td><td>1</td><td>1</td><td colspan="2">CID</td><td></td></tr>
<tr><td colspan="5">CID</td><td>(If 16-bit CID is used)</td></tr>
<tr><td colspan="3">CID</td><td>M</td><td>Link Seq = 0</td><td></td></tr>
<tr><td colspan="5">Original Sequence number</td><td></td></tr>
<tr><td colspan="5">Original Timestamp</td><td></td></tr>
<tr><td colspan="5">Original Identification</td><td></td></tr>
<tr><td colspan="5">Payload</td><td></td></tr>
</table>

Figure 3-10: Compressed REFERENCE_HEADER packet format

In the ideal case, where every consecutive packet can be compressed with maximum efficiency, CRTP Header Compression reduces overhead by 90% with the UDP checksum enabled, and 95% in best-case compression per packet (see Figure 3-11) compare to full header size. This reduces the bandwidth requirement significantly and also achieves better packet loss rates over packet-switched wireless channels (Figure 3-13, page 74). The details of the simulation scenario are given in section 3.6.3.

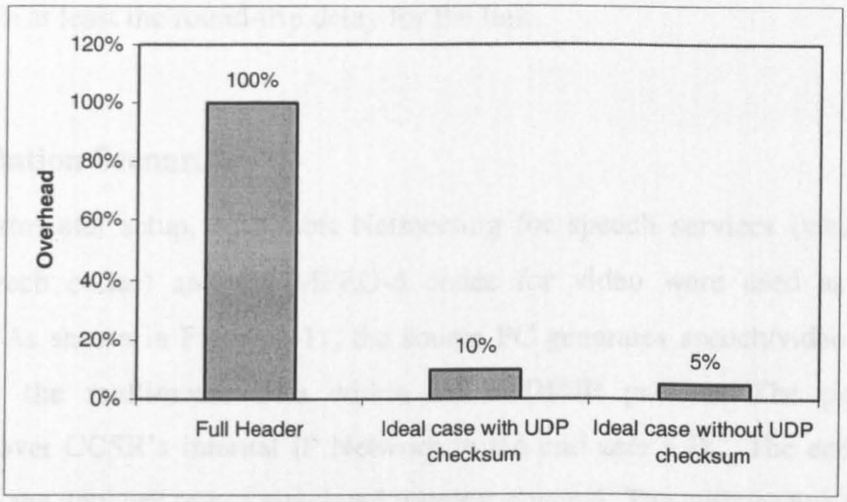


Figure 3-11: Efficiency of CRTP Header Compression

3.6.2.5 De-compressor Feedback: Context State

Since wireless channels have very high bit error rates (BER) due to shadows, multipath fading, and continuous mobility, the predictive coding used by the CRTP approach has a very serious weakness; its fragility when subjected to channel errors. If a packet gets lost or corrupted and cannot be decompressed, the context of the de-compressor loses synchronisation with the compressor, since it is not being updated correctly. As a result of this, all subsequent headers will either be incorrectly reconstructed or discarded, which will cause the context to be invalidated. The resulting out-of-date context will have to be replaced by a new uncompressed, and preferably undamaged, header to bring the de-compressor context in to sync with the compressor. This is achieved by means of a feedback channel from the receiver to the compressor, indicating the most recently correctly decoded packet. The de-compressor sends a NEGATIVE ACKNOWLEDGMENT, called "*Context State*", which indicates a special packet that carries the CID and sequence number of the packet that has been corrupted or lost. This means that the roundtrip time over the link determines the speed of the repair mechanism. In the simulations, it is assumed that the context state messages sent via the feedback channel never get damaged.

Sending full-headers or compressed_UDP headers repairs the damaged context at the de-compressor. The compressor relies on the CONTEXT_STATE massage from the de-compressor, to send these headers. This means that the time taken by the repair mechanism is at least the round-trip delay for the link.

3.6.3 Simulation Scenario

In the experimental setup, Microsoft Netmeeting for speech services (which uses the G.723.1 speech codec) and the MPEG-4 codec for video were used as the source application. As shown in Figure 3-11, the source PC generates speech/video frames and encapsulates the multimedia data within RTP/UDP/IP packets. The packets were transmitted over CCSR's internal IP Network to the end user's PC. The end user PC is connected to the network over a simulated wireless channel. The cellular link is simulated using a GPRS channel model. Header compression is applied over the simulated wireless link.

Typical BERs for cellular systems are generally in a range between 10^{-2} and 10^{-6}. The most recently standardised speech codecs have inbuilt error resilience techniques, which make them robust enough to deliver acceptable speech quality over cellular channels whose BER can fall to 10^{-2}.

In the simulations carried out, the round trip time over the link varied between 120-160 ms (the time for 6-8 speech frames). This RTT is representative of the delays in the packet-switched access networks. In total, 9600 packets are transmitted. Packets can be damaged due to the errors over the wireless link. In the experiments, a packet is considered lost if it is not passed up to the application layer. In this study, this can be due to two different reasons:

a) A bit error might damage the compressed header and the de-compressor cannot reconstruct the original header.
b) The context of the decompressor may lose synchronisation with the context of the compressor, thereby possibly preventing subsequent packets from being decompressed. This can happen even if the compressed header is error free.

NOTE: The CRTP scheme's performance measurements focused on the compression efficiency and robustness. Internet congestion and the consequent late arrival of packets are not represented in this chapter study; otherwise the results would not be comparable to the results published by the IETF committee [81]. It is assumed that all packets arrive on time. Also, a packet with errors in the payload is not regarded as lost either.

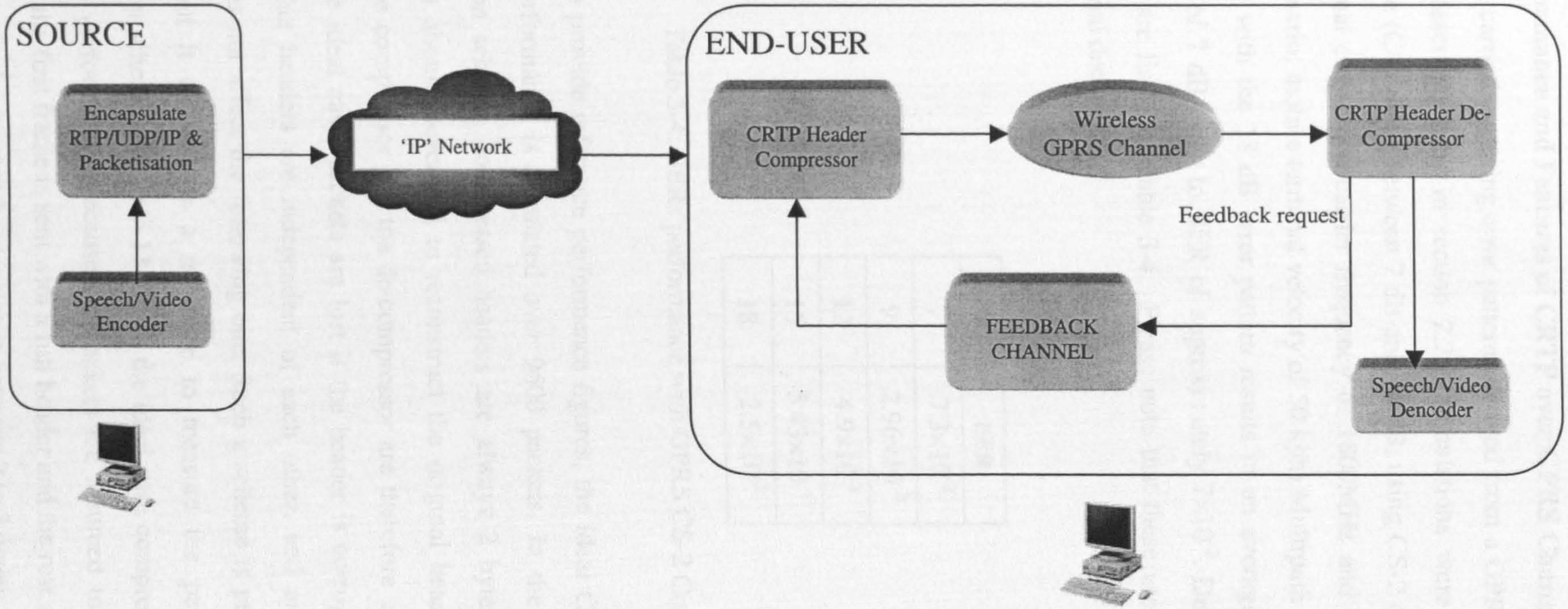


Figure 3-12: Simulation Scenario

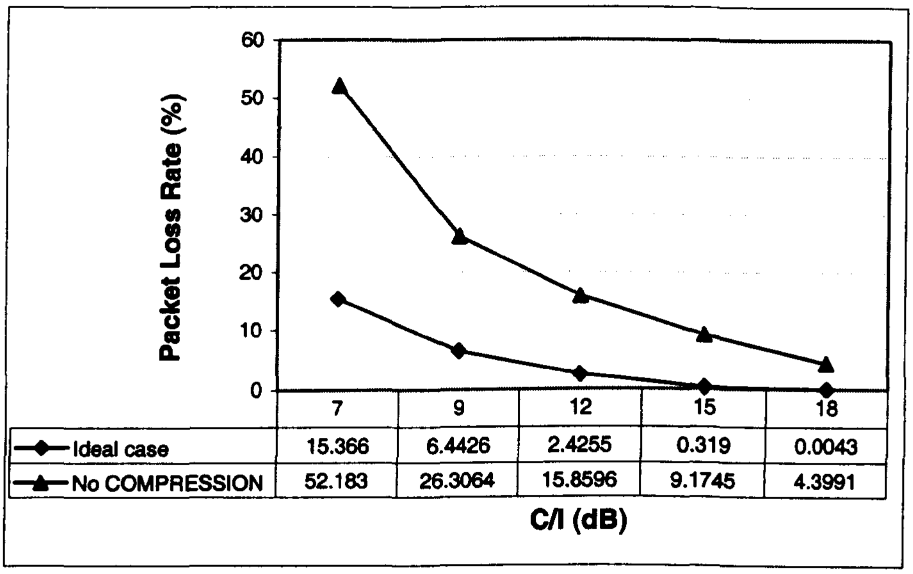
3.6.4 Performance and Features of CRTP over GPRS Channels

Tests were carried out using error patterns derived from a GPRS channel model [3][21], which is described fully in section 2.2.2. Simulations were performed at carrier-to-interference (C/I) ratios between 7 dB and 18 dB, using CS-2 channel coding (a rate 2/3 convolutional code) at a carrier frequency of 1800MHz and using the TU50 (Typical Urban Scenario, mobile terminal velocity of 50 kph) Multipath model as specified in [3]. Corruption with the 18 dB error pattern results in an average BER of around 2×10^{-5}, where C/I of 7 dB leads to BER of approximately 7×10^{-2}. Details of the other channel conditions are listed in Table 3-4. Please note that these are the BER presented after convolutional decoding.

C/I (dB)	BER
7	7.3×10^{-2}
9	2.96×10^{-2}
12	4.9×10^{-3}
15	5.43×10^{-4}
18	2.5×10^{-5}

Table 3-4: BERs performance with GPRS CS-2 Coding Scheme

In order to provide reference performance figures, the ideal CRTP header compression scheme performance is simulated over 9600 packets. In the reference CRTP header compression scheme, compressed headers are always 2 bytes in length, and the de-compressor should never fail to reconstruct the original header from the compressed header. The compressor and the de-compressor are therefore always in sync. However, even in the ideal case, packets are lost if the header is corrupted. Effectively the it is assumed that headers are independent of each other, and any corrupted compressed header does not affect the following one. Such a scheme is probably not achievable in practice, but it is used as a reference to measure the performance of the tested compression scheme. Figure 3-13 shows the ideal case compression scheme packet loss rate (PLR) performance. Because the packets are assumed to be independent of each other, only the first frame is sent with a full header and the rest are sent using compressed headers. Hence, the average header size is between 2 to 3 octets.



	7	9	12	15	18
—◆— Ideal case	15.366	6.4426	2.4255	0.319	0.0043
—▲— No COMPRESSION	52.183	26.3064	15.8596	9.1745	4.3991

Figure 3-13: PLR for Ideal Case CRTP compression Scheme and PLR when no compression is employed

Figure 3-13 shows that packet transmissions using header compression reduce the PLR significantly compared with packet transmission without header compression. This is because the packets with full headers are larger than the packets with compressed headers, so it is more likely that a packet with a full header is corrupted than the packet with a compressed header under identical channel conditions.

Using standard CRTP compression schemes the de-compressor always reconstructs the header information by using the last reconstructed header. In this work, an extra packet format, known as a REFERENCE_HEADER, is introduced (see section 3.5.2.4). Figure 3-14, demonstrates that CRTP header compression with the proposed REFERENCE_HEADER format, improves the PLR performance of the CRTP header compression scheme. The packet loss rate at a C/I of 7dB, which is equivalent to a BER of 7x10^{-2}, is reduced from 40.6% to 32.7%. The improvement compared with the "no _compression" application is 19.4 percent, a reduction from 52.1% to 32.7%.

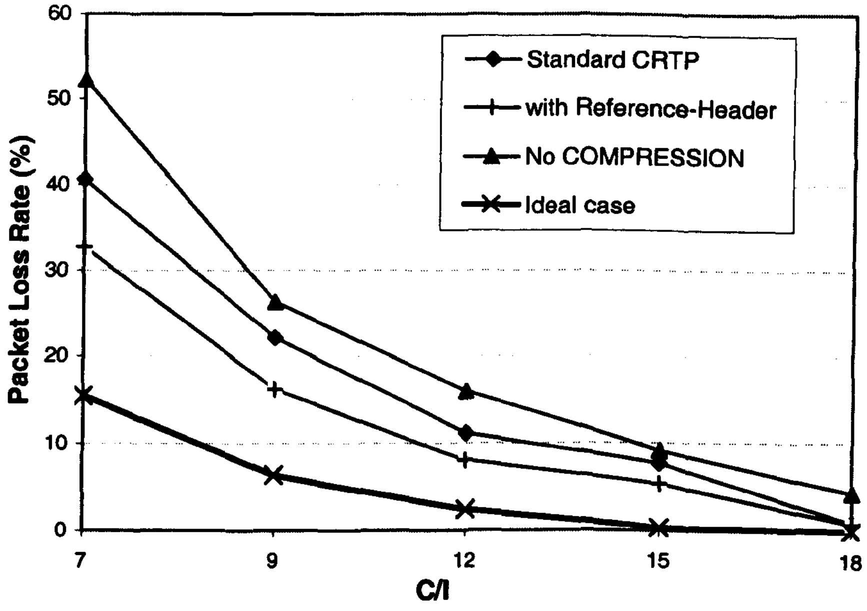


Figure 3-14: Packet Loss Rate of CRTP with Reference_header against standard and ideal case CRTP when GPRS CS-2 is used

Figure 3-15 shows that including the REFERENCE_HEADER format increases the average overhead requirement per packet by about 13%, requiring 2.4 octets with a C/I of 7 dB. As the compression is done based on the REFERENCE_HEADER, the de-compressor only uses reference header to reconstruct the original header. Therefore, this requires a greater number of bits that increases average header size.

Even though reference-based compression improves the packet loss rate performance, the results presented in Figure 3-14, show that there is still a significant gap with the ideal scheme. Including REFERENCE_HEADER in the CRTP compression scheme enhances the PLR performance, but at the same time it increases the average header size.

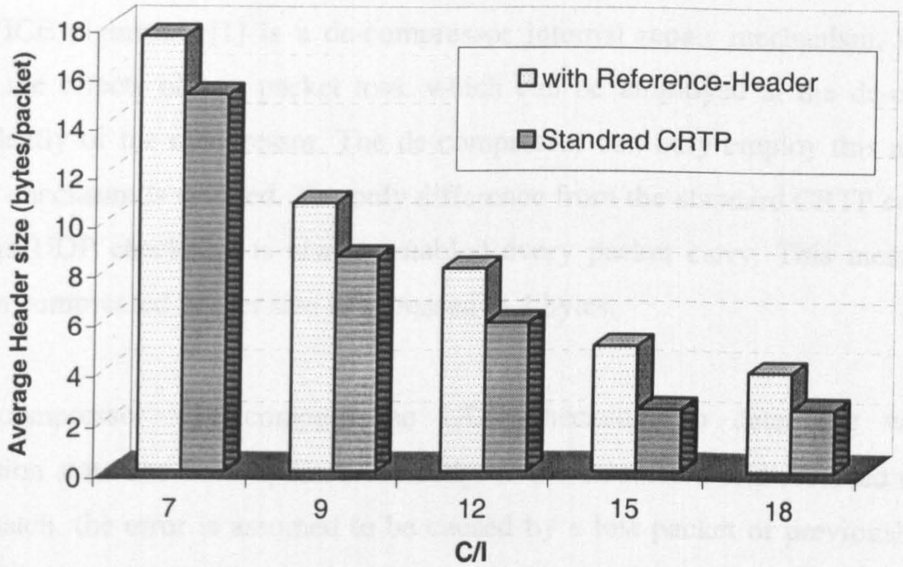


Figure 3-15: Average header size when GPRS CS-2 is used

3.6.5 Resilience Techniques: with/without feedback

The service quality of data communications using IP packet-switched networks without mechanisms such as RSVP, DiffServ, is based on a best-effort approach. Best-effort services do not give any guarantees regarding delay. Delay is not critical for certain applications, such as e-mail, fax etc., i.e. non-real time services. For these services, the desired service quality can be achieved by using a feedback channel that sends ACK/NACK information from the de-compressor to the compressor. However, delay is one of the most critical problems when using packet-switched networks for real-time services such as interactive conversations. Minimising end-to-end delay for real-time applications is critical. It can be more tolerable for one-way audio/video communications, such as real-time streaming, since the receiver can adapt to the variance in delay.

As is shown in Figure 3-14, the damage to packet headers significantly decreases the efficiency of the standard CRTP header compression algorithm. An effective header compression scheme should be robust enough to minimise the effect of errors on the compression performance and packet loss. In the following sections, three different resilience schemes are introduced to increase the efficiency of CRTP header compression.

3.6.5.1 TWICE algorithm

The TWICE algorithm [1] is a de-compressor internal repair mechanism, intended to mitigate the effects of any packet loss, which can be employed at the de-compressor, independently of the compressor. The de-compressor can only employ this algorithm if the UDP checksum is enabled. The only difference from the standard CRTP compression scheme is UDP checksum is always enabled every packet carry. This means that the minimum compressed header size is increased to 4 bytes.

The de-compressor will compute the UDP checksum to determine whether its compression state has been updated correctly. If the calculated and received checksums do not match, the error is assumed to be caused by a lost packet or previously received corrupted packet, and it is also assumed that the header information could not be updated properly at the de-compressor. As it is highly likely that the lost packet's header information contained the same delta as the current packet, the delta values of the individual fields are added to the last reconstructed header. This is a way to repair the context without having to wait for feedback over a link. Note that the standard CRTP header compression scheme uses the UDP checksum. As this checksum covers the entire payload rather than just the header and any error on the payload would cause the UDP Checksum to fail, therefore the UDP Checksum cannot be used for this purpose. Instead a 16-bit cyclic redundancy checksum (CRC) [40] is introduced that covers the original header, and is used to validate the reconstructed header at the de-compressor. The polynomial of CRC-16 is shown in the following:

$$C(x) = x^{16} + x^{15} + x^{2} + 1$$

The 16-bit checksum is capable of detecting error bursts of up to 16 bits in length. The checksum number is represented as a 16-bit unsigned number, encompassing the range 0 ... 65535, meaning there is one chance in 65535 of an error not being detected, where two different headers have the same checksum. By de-compressing and computing this 16-bit CRC again, the de-compressor checks if the repair has succeeded or if the delta should be applied once more.

The TWICE algorithm was tested using the G.729b speech codec. This speech codec generates packets with payloads of fixed size (10 octets), which represents the smallest reasonable payload size. Packets are produced at a rate of 100 per second.

The plots in Figure 3-16 present the packet loss rate performance for standard CRTP, CRTP with TWICE, and the Ideal scheme. The figure shows that repairing the context by using the Twice scheme increases the packet loss rate (PLR) performance slightly. Performance though, is still significantly inferior to the ideal case. At 9 dB C/I, which corresponds to a residual bit error rate (BER) of approximately 3×10^{-3}, the packet loss rate (PLR) is approximately 7×10^{-2} for Ideal case compression scheme which is explained in section 3.6.2.4, about 1.9×10^{-1} for CRTP with Twice, and 2.2×10^{-1} for the standard CRTP scheme.

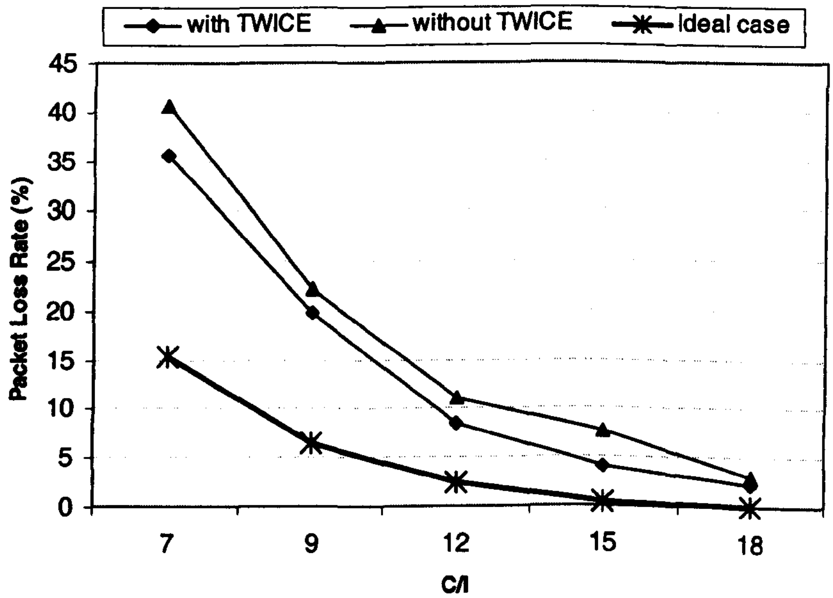


Figure 3-16: Packet Loss Rate for CRTP, CRTP with Twice and Ideal case

However, the improvement in robustness is not matched by an improvement in the efficiency performance. Figure 3-17 presents the required average header size, when TWICE error resilience algorithm is in used. Since extra 2 bytes are employed on every compressed header, the average header size is increased from 7.015 to 8.64 bytes, which is about 23 %, at 9 dB C/I.

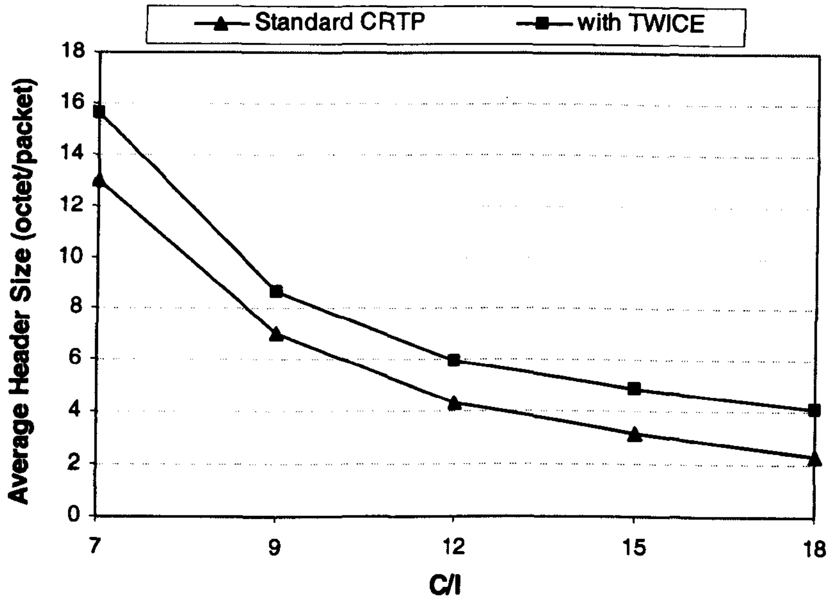


Figure 3-17: Average Header size for CRTP and CRTP with Twice

3.6.5.2 Slow Start Update Scheme (SSUS) with feedback channel

The *Slow Start Update Scheme* is a new resilience scheme for CRTP, which is proposed by the author. This is implemented at the encoder, and is used when wireless channels are very error prone and also when the end-to-end delay is higher than the packet's inter-arrival period. It allows the decompresser to recover quickly from the loss of a full header that would have changed the corrupted compression state. The full headers are sent periodically with an exponentially increasing period following a change in the compression state. This technique minimises the exchange of messages between compressor and de-compressor, a property which is very important in a mobile environment, since a mobile user is subject to periods of fading, or high bit error rates. The fade period can be up to a few seconds in length causing any transmission to be delayed or lost.

If the round-trip delay is larger than the inter-packet spacing, which is the gap between two consecutive full headers, the de-compressor will have to discard a number of outdated headers, before an entire (uncompressed) header arrives. If error detection is employed, an exponential refresh period may be used. When a new uncompressed header is requested, the encoder can start a slow update scheme. The slow update scheme sends an uncompressed header, followed by a single compressed header, followed by another uncompressed header. The period between header refreshes then increases exponentially to 2, 4, 8 etc. until a value of 255 is reached. If errors are detected at the de-compressor, the period counter is calculated based on delay as shown in equation 3.W. This exponential refresh period allows the algorithm to adapt to varying channel conditions.

F = full header,

C = compressed header

C_PERIOD = number of compressed header between two full header

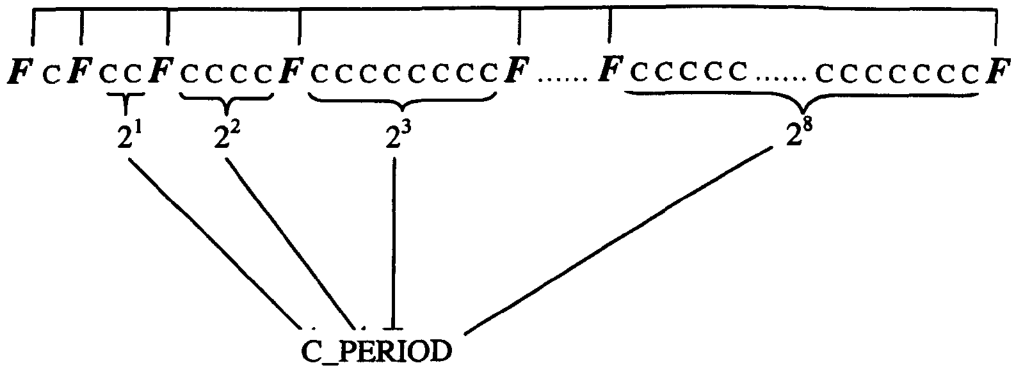

F c F c c F c c c c F c c c c c c c c F ...... F c c c c c ...... c c c c c c c F
2^1
2^2
2^3
2^8
C_PERIOD

Figure 3-18: SSUS algorithms packet transmission pattern

If Compressed Header loss/corrupted:

- Delay ≤ ((C_Period)—(# of Pr)) (YES)
- Feedback (NACK)
- C_Period = 2^n (where n = floor [(log2 (delay)])

Figure 3-18 shows how packets are sent. C_PERIOD represents the number of compressed headers (C) between two full headers (F). It keeps track of how many compressed headers have been sent between full headers. When the compression state changes or if a request is received from the de-compressor, a full header is sent and the C_PERIOD is set to one. C_PERIOD is doubled each time a full header is sent during compression slow-start.

Figure 3-19 shows the PLR for this scheme. It can be seen that the packet loss rate is reduced by about 27 %, from 40 % for standard CRTP to 29.55 % at a C/I of 7 dB. However, it is still much higher than for the Ideal case. In the simulation, it is assumed that the refreshed header (F) is not damaged over the mobile channel. This can be achievable by using extra protection. Over an error prone environment, the header is updated more often. Since the refresh header is larger, it is more likely that packets are damaged.

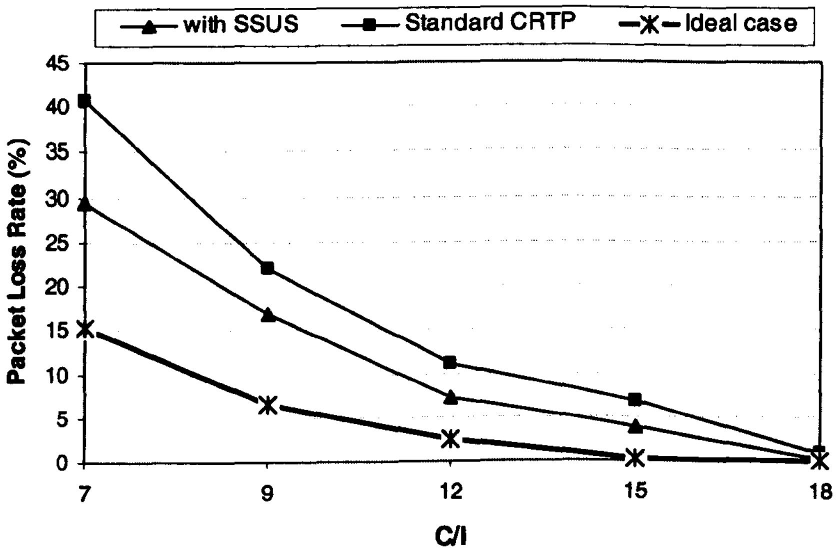


Figure 3-19: PLR of SSUS against the standard CRTP and Ideal case CRTP

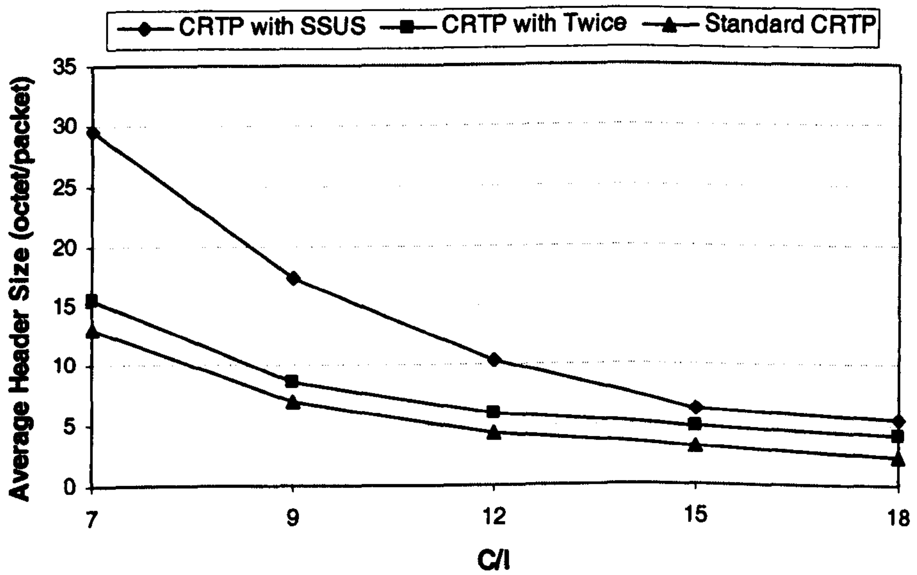


Figure 3-20: Average Header Size CRTP with SSUS and TWICE against standard CRTP

Figure 3-20 shows the average header size per packet when SSUS error resilience algorithm is implemented. Unfortunately, the price to pay for the Packet Loss Rate improvement with SSUS scheme is extra overhead. At 7 dB C/I, the average header is increased by about 17 octets compared to standard CRTP usage. It is clear in the Figure 3-20 that, as channel conditions improve, fewer refresh headers are needed. Therefore, the overhead that is required for each packet is reduced.

3.6.5.3 Fixed-Period Refresh Scheme

All of the previously introduced error resilience techniques require a feedback channel to deliver their improved performance. However, techniques that require an exchange of messages cannot be used over channels where there is no feedback channel, such as direct-broadcast satellite channels or cable TV systems. Such techniques are also hard to adapt to multicast applications. The SSUS scheme minimises the use of the feedback channel, but still requires a feedback channel to achieve the best packet loss rate (PLR) performance. The *Fixed Period-refresh* scheme is presented as an alternative technique for these kinds of applications. The Fixed Period-refresh scheme aims to update the context more regularly than any of the request-based schemes. Fixed-period refreshes are sent at shorter intervals than the link round trip time regardless of the channel conditions. It fixes a period between two full headers. Between these two headers, it sends compressed packets. The full headers always are sent with same fixed interval.

The G.729b speech codec was used to compress speech at an output bit rate of 8kbit/s. The frame length is 10 ms, which results in 80-bits/frame. The frames are sent with one-frame-to-one RTP packet mapping, which produces a packet rate of 100 packets per sec.

In the simulations carried out, the link round-trip time is between 120-160 ms (12-16 frame), which means that with standard request-based CRTP, a single discarded packet causes a further 12-16 additional frames to be lost due to context damage. By using fixed-period refresh, the full header is sent every 4th and every 6th packet respectively, in two different simulations. The compressed-header is used for the rest of the packets. 9600 packets were used in the simulations.

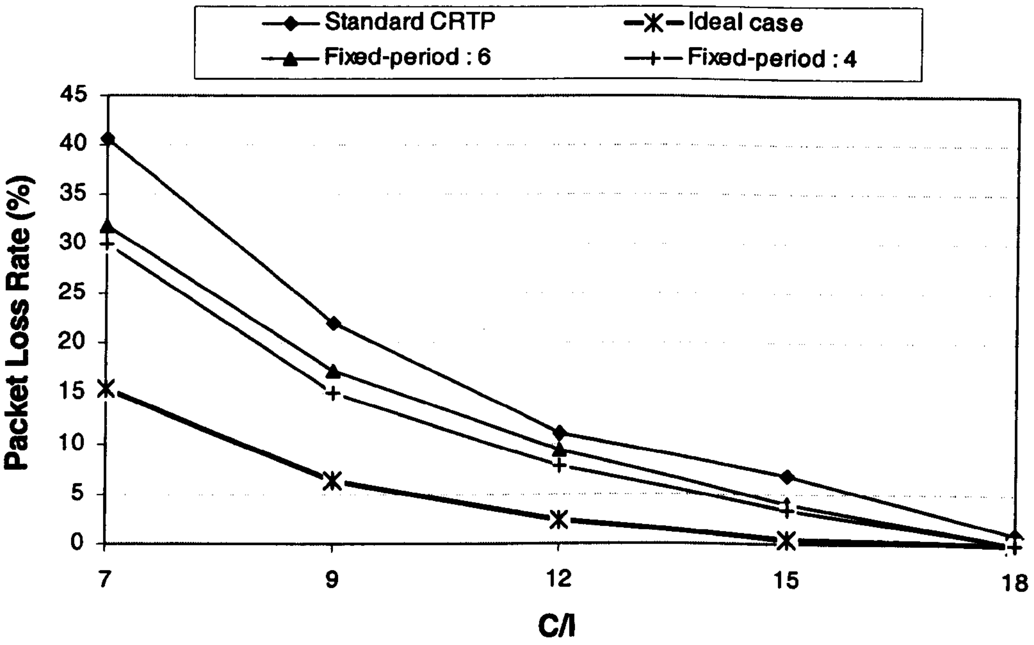


Figure 3-21: PLR for CRTP fixed_period refreshes against standard and Ideal case CRTP

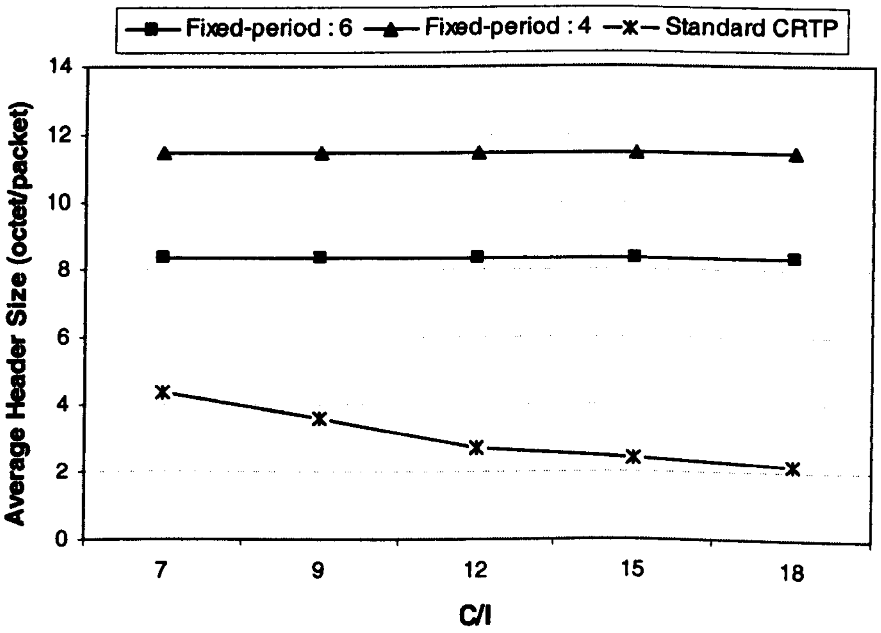


Figure 3-22: Average Header Size CRTP with fixed_period refreshes against standard CRTP

As shown in Figure 3-21, the fixed-period refresh scheme reduces the Packet Loss Rate values by 8.78 % and 10.69 % with periods of 4 and 6 respectively at 7 dB of C/I, compared with the standard request-based CRTP. However, it is still much higher than the Ideal scheme. Unfortunately, average header size is also increased by a factor between 2 and 3, as shown in Figure 3-22.

It is clear that the fixed-period refreshing scheme improves the packet loss rate (PLR) over links with a round-trip time of 120-160 ms. With even longer RTT's, fixed-period refresh scheme is more applicable for the packet loss rate (PLR) improvement. However, it is much more costly in terms of header size than standard request-based CRTP.

3.7 Conclusion

In this chapter, one of the most critical aspects of transmitting real-time multimedia information over packet-switched mobile networks, which is the significantly large RTP/UDP/IP overhead, is discussed. In addition, the characteristics of the throughput of mobile channels and their error characteristics are examined, and are seen to be time-varying. This Chapter has examined the characteristics of the header fields in detail, the way they change during multimedia transmission, and the advantages of the header compression. To examine the characteristics of the header fields, header 'capture and analyser' program is implemented in the window environments.

The Compressing RTP/UDP/IP (CRTP) header compression technique is the first algorithm, which is used to compress the real-time protocol stacks over mobile links. It was proposed in 1998 by the IETF standard committee, audio/video transport group. In this chapter, this algorithm has been implemented by the author, based on the IETF standard. The performance of this scheme is evaluated over simulated GPRS mobile channel. The results are presented in Figure 3-13, with the ideal case outcome of CRTP. It is clear from the Figure 3-10 that CRTP does not perform well over mobile channels. The major cause of CRTPs bad performance is that many packets are discarded due to context damage. The performance was improved by using REFERENCE_HEADER format together with the original format. This format is proposed by the author. It is used to increase robustness of transmission. By using the REFERENCE-HEADER packet, any error propagation due to the context damage can be stopped. The results, demonstrated in

Table 3-5, shows that integrating the reference_header format increases the packet loss rate performance, however it also increases the required average header size, see Figure 3-15

This chapter has also introduced three different error resilience techniques to improve the performance of the CRTP algorithm. Two of these techniques, TWICE and periodic refreshes, were proposed in the IETF. The third scheme, SSUS is proposed by the author.

TWICE scheme is a de-compressor internal mechanism, intended to mitigate the effects of any packet loss. It requires the UDP checksum to be enabled to verify its repair attempts, which cost two extra octets of the header for every packet. The simulation results are presented in table 3-5.

Periodic refresh was proposed for simplex links, where the feedback channel is not available, and links with very high delay. The periodic refresh method updates the context with fixed period. It does not require context updates request from the de-compressor. The average header sizes do not vary according to the channel conditions.

C/I (dB)	7	9	12	15	18
CRTP Ideal Case	15.4 %	6.52 %	2.42 %	0.32 %	0.0 %
CRTP with SSUS	29.6 %	16.9 %	7.2 %	3.8 %	1.7 %
CRTP with TWICE	35.5 %	19.7 %	8.4 %	4 %	2.12 %
CRTP fixed_period refreshes=4	29.95 %	15.02 %	7.73 %	3.47 %	0.08 %
CRTP fixed_period refreshes=6	31.63 %	17.1 %	9.1 %	4.75 %	0.1 %
Reference Header	32.7 %	16.1 %	8.1 %	5.4 %	0.83 %
Standard CRTP	40.6 %	22 %	11 %	7.7 %	2.9 %

Table 3-5: Overall Packet Loss Rate Comparisons

The final error resilient scheme is *SSUS*, which was proposed in this Chapter. It is employed at the encoder, and is based on feedback of the channel conditions, including end-to-end delay. The encoder decides when full headers need to be sent. The performance of the packet loss rate results shown in Table 3-5 shows that the SSUS method improves the packet loss rate performance against standard CRTP, however it increases the average header size significantly (see Figure 3-22). However, it is the most costly method regarding to the bandwidth.

As shown in Table 3-5, all the error resilience schemes used in the simulations reduced the packet loss rate. However these schemes are costly in terms of header size. The improvements in the packet loss rate that have been achieved by employing these resilience methods are not sufficient for mobile communications. It can be summarized that CRTP does not function adequately over lossy links, especially those with long roundtrip times. With all the error resilience methods that have been tried, the performance of CRTP is still worse than the Ideal case in terms of the packet loss rate as well as the average header size.

Chapter 4

4 *R*obust *C*hecksum-based header *CO*mpression (ROCCO) & Enhancement

4.1 Introduction

CRTP with a feedback channel, including the different resilience schemes described in chapter 3, can achieve considerably better performance than both raw (uncompressed) headers. However, chapter 3 showed that a deterioration in channel quality results in a considerable increase in the packet loss rate, and in an increase in the packet overhead caused by the requirement to re-initialise the context at the receiver. For this reason, an alternative scheme has been developed and published as an IETF draft in 2000 called robust checksum-based header compression (ROCCO) [6], which is a RFC draft now and is called ROHC (**RO**bust **H**eader **C**ompression). It had been implemented as part of this study.

Its basic framework is supplemented with a set of compression profiles, where each compression profile provides the exact details on how a packet is to be compressed and de-compressed. The use of compression profiles allows optimal compression performance for different traffic characteristics (such as audio and video) and for different channel conditions. In ROCCO, the compressed header carries a following 3-bit, 7-bit or 8-bit checksum (CRC), which is computed for the headers that are to be compressed. The choice of CRC is based on the formats which are used for different channel conditions. This provides a reliable way of detecting whether decompression and context repair has succeeded (see Figure 4-1). The compressed ROCCO header also contains a code that specifies how the header fields have changed. This information allows the decompressor to carry out a local repair of the header context, which allows it to perform better than CRTP when several consecutive packets have been lost or corrupted. This local repair functionality reduces the need for the decompressor to request a new (uncompressed) set of headers from the transmitter. This approach has been shown to achieve improved performance; however the complexity issue is not discussed by the scheme designers.

3-bit CRC polynomial:

$$C(x) = x^3 + x + 1$$

7-bit CRC polynomial:

$$C(x) = x^7 + x^6 + x^3 + x^2 + x + 1$$

8-bit CRC polynomial:

$$C(x) = x^8 + x^2 + x + 1$$

In this Chapter, the implementation of ROCCO and its performance is investigated, and the results presented in details. In addition, two resilience techniques are proposed by the author. The first one is Adaptive_Reference Update Scheme (**ARUS**) which minimises/stops the risk of context damage propagation within the ROCCO algorithm. With this method, a new compressed packet format is introduced, which is called compressed *adaptive_reference header*. The second resilience method is called prioritisation, which is built on top of the ARUS method and increases the performance improvement in terms of Packet Loss Rate (PLR). These are tested with various standard speech codecs, which have different characteristics, and the MPEG-4 video codec. Two resilience techniques are proposed in this work, which are shown to improve the performance of ROCCO.

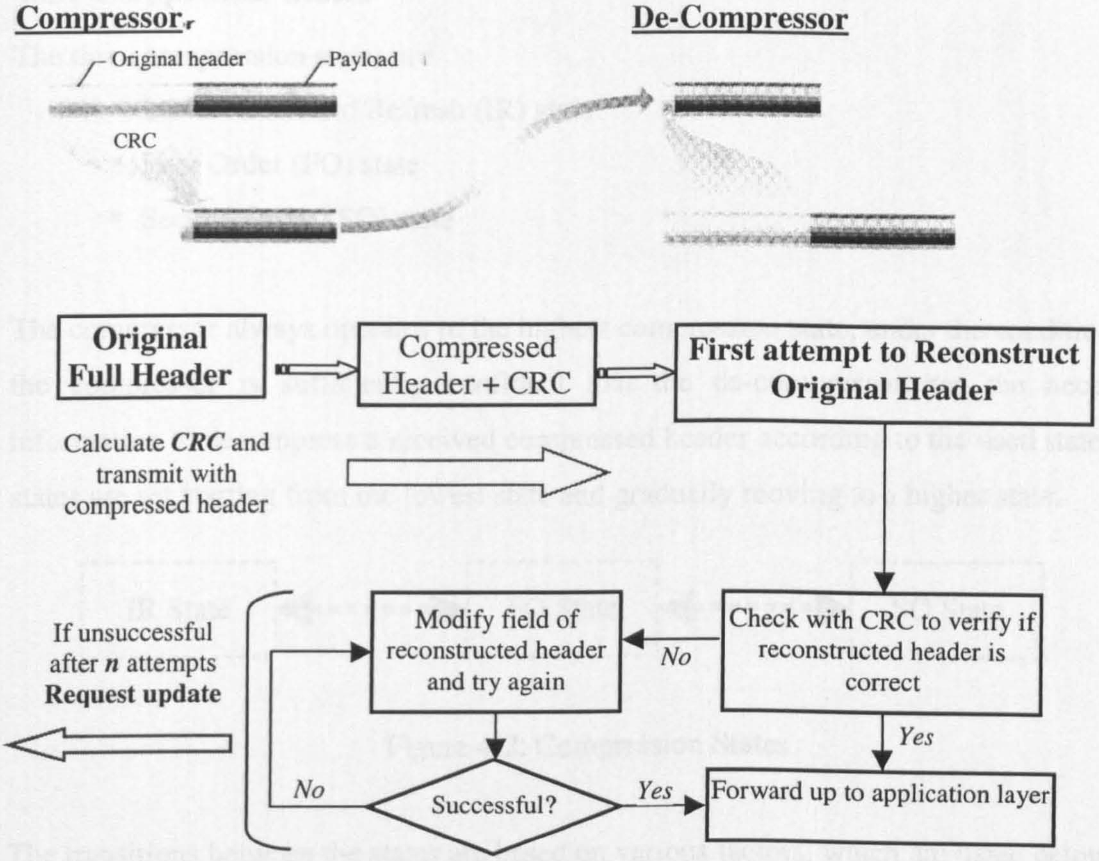

Compressor
De-Compressor
Original header
Payload
CRC
Original Full Header
Compressed Header + CRC
First attempt to Reconstruct Original Header
Calculate CRC and transmit with compressed header
If unsuccessful after n attempts Request update
Modify field of reconstructed header and try again
No
Check with CRC to verify if reconstructed header is correct
Yes
No
Successful?
Yes
Forward up to application layer

Figure 4-1: ROCCO- the robust checksum-based header compression scheme

4.2 Compressing and Decompressing States

Header compression with ROCCO is not as simple as with CRTP scheme. The compressor and de-compressor need to share some information, which should be provided at both ends in advance [6] [95]. The interactions help to achieve more efficient and more robust compression between the two ends. The compressor and de-compressor have three states, which are related to each other, though the precise meanings of the states are slightly different for the two parties. These states are classified based on the capability and condition of channel, and the irregularities of the header fields. Both compressor and de-compressor start in the lowest compression state (maximum header), which is Initialisation and Refresh (IR) state and gradually transit to higher states (minimum compressed header), which is Second Order (SO) state. The following subsequent sections present an overview of the compressor and de-compressor and their corresponding states.

4.2.1 Compressor States

The three compression states are

- Initialisation and Refresh (IR) state
- First Order (FO) state
- Second Order (SO) state

The compressor always operates in the highest compression state, under the condition that the compressor is sufficiently confident that the de-compressor has the necessary information to decompress a received compressed header according to the used state. The states are set starting from the lowest state and gradually moving to a higher state.

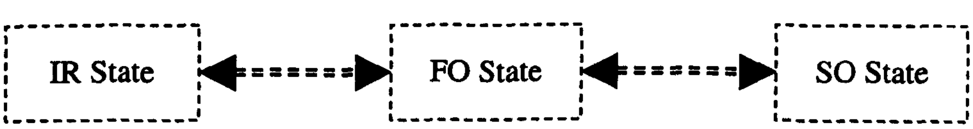


Figure 4-2: Compression States

The transitions between the states are based on various factors, which are listed below:

- Variations in consecutive packet headers
- Positive feedback from the de-compressor, which is known as Acknowledgment, "ACK"
- Negative feedback from the de-compressor, which is known as Negative ACK, "NACK"
- Periodic timeouts (where a feedback channel is not possible, i.e., over simplex channels)

4.2.1.1 Initialisation and Refresh (IR) States

As indicated by its name, the IR state is used to initialise the static (inactive and defined) parts of the context at the de-compressor, or to recover after failure. In this state, the compressor sends complete header information, regardless of whether they are static or non-static, plus some additional information (CID and compression profile code), which helps the de-compressor to initialise the context. The format is shown in section 4.8. The compressor stays in the IR state until it is confident that the de-compressor has received the static information correctly.

4.2.1.2 First Order (FO) State and Packet type

The main purpose of the FO state is to efficiently communicate irregularities in the packet stream. When operating in this state, the compressor rarely sends information about all dynamic fields and the information sent is usually compressed at least partially [6]. Only a few static fields can be updated. The format is shown in section 4.9.

4.2.1.3 Second Order (SO) States and Packet Type

When the transmission is in the SO state, the compression is optimal. This state is entered when the header is completely predictable for a given SN (RTP sequence number), and the compressor is confident that the de-compressor has obtained all required parameters required to reconstruct the original header. So, it is clear that successful de-compression depends on information sent in the preceding FO state packets being successfully received by the de-compressor. When header information no longer conforms to a uniform pattern and cannot be independently compressed on the basis of previous context information, the compressor leaves this state and goes back to the FO state.

4.2.2 Decompression States

The de-compressor operates in three states, which are;

- No Context State
- Static Context State
- Full Context State

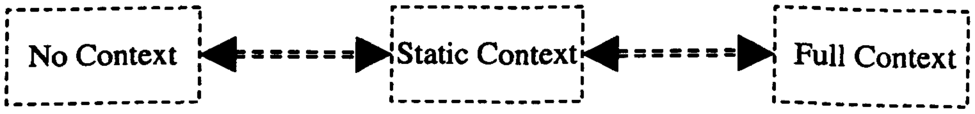


Figure 4-3: Decompression States

Similarly to the compressor, the de-compressor starts in its lowest compression state, which is the "No Context" state, and moves to higher states when appropriate. The de-compressor state transitions are based on successful decompression of received packets. Once the received compressed header has been de-compressed correctly, the de-compressor can move all the way to the "Full Context". If decompression of several packets fails, it switches back to the lower state. If this failure continues, the de-compressor goes all the way back to the "No Context" State. The de-compressor is only required to acknowledge the compressor when it switches to the lowest state.

4.3 Mobile Channel Status

The status of the mobile channel plays a very significant role in packet communications. Channels have different characteristics. For example, in some links a feedback channel cannot be used. The ROCCO algorithm takes into consideration all kind of situations, operating in three different modes. The modes are categorised based on the channel status. In the following subsections, the modes are explained in details. The states, explained in the previous section, and the various modes are orthogonal to each other. The state concept is the same for all mode operations, while the actual mode controls the logic of state transitions, and what actions to perform in each state. These modes are:

- Unidirectional Mode
- Bidirectional Optimistic Mode
- Bidirectional Reliable Mode

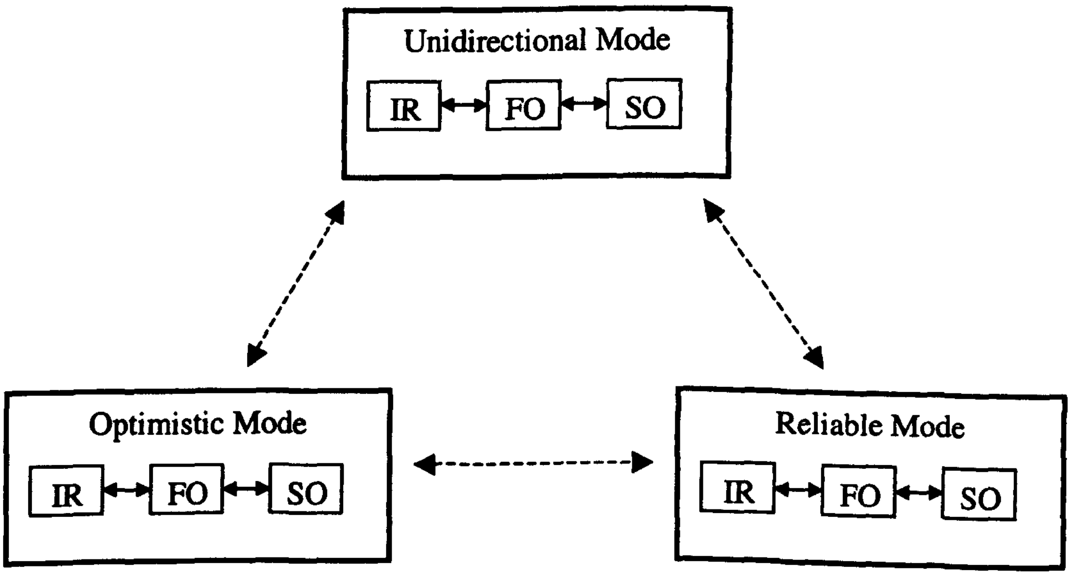


Figure 4-4: Three Mode and states

4.3.1 Unidirectional Mode (U-Mode)

Unidirectional Mode is designed for links where a return path from de-compressor to compressor is unavailable or undesirable. When Unidirectional Mode is in operation, packets are only sent in one direction, from compressor to de-compressor. This mode is also used at the beginning of packet transmission in any communication session. As soon as a packet is received, the de-compressor can reply with a feedback packet indicating that a mode transition is desired. If there is no feedback, the compressor stays in U-mode.

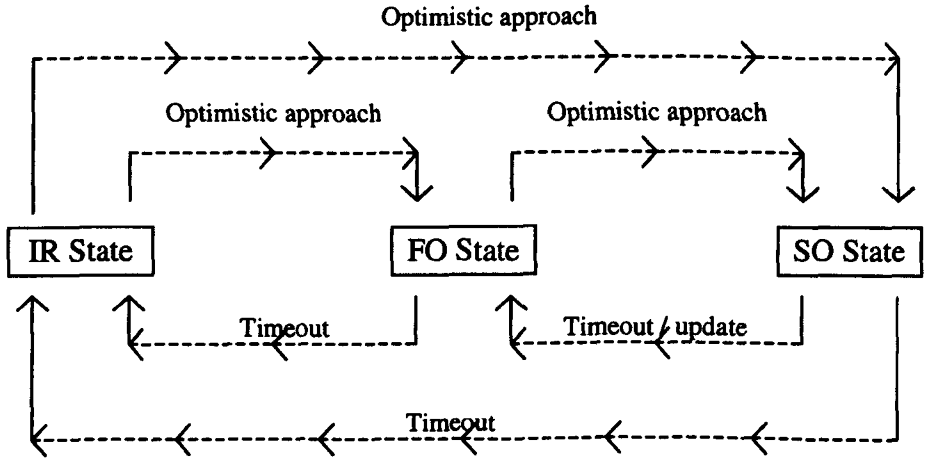


Figure 4-5: Unidirectional Mode and Transition States

Due to the lack of feedback for initiation of error recovery, compression in the U-mode is less efficient. This has a negative impact on packet loss or error propagation due to context damage. The transitions between compressor states are performed based on periodic timeouts, and irregularities in the header field changes in the compressed packet stream. Figure 4-5 shows the details of the transitions between states and compression logic.

4.3.2 Bidirectional Optimistic Mode (O-Mode)

The Bidirectional Optimistic mode requires a feedback channel. The feedback channel is used for error recovery request and acknowledgment of correctly received significant context updates. This mode aims to maximise compression efficiency and minimise the propagation loss by sparse usage of the feedback channel. Figure 4-6 shows the details of the transitions between states and compression logic.

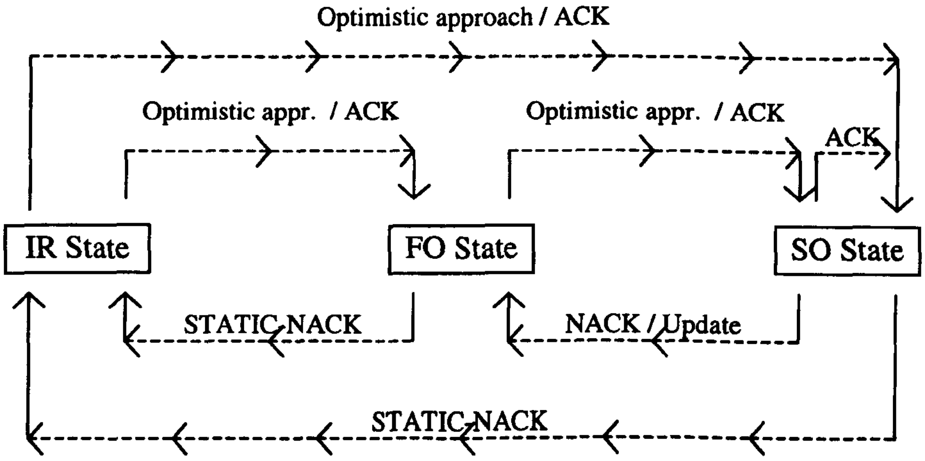


Figure 4-6: Bi-directional Optimistic Mode and transition states

This mode reduces the number of damaged headers delivered to the upper layers due to the residual error or context invalidation.

4.3.3 Bidirectional Reliable Mode (R-Mode)

Bidirectional Reliable mode is similar to Optimistic mode except for more intensive use of the feedback channel. Generally, only a CRC is employed here. This CRC is calculated over the original header, which is compared with the CRC calculated over the reconstructed header by the de-compressor.

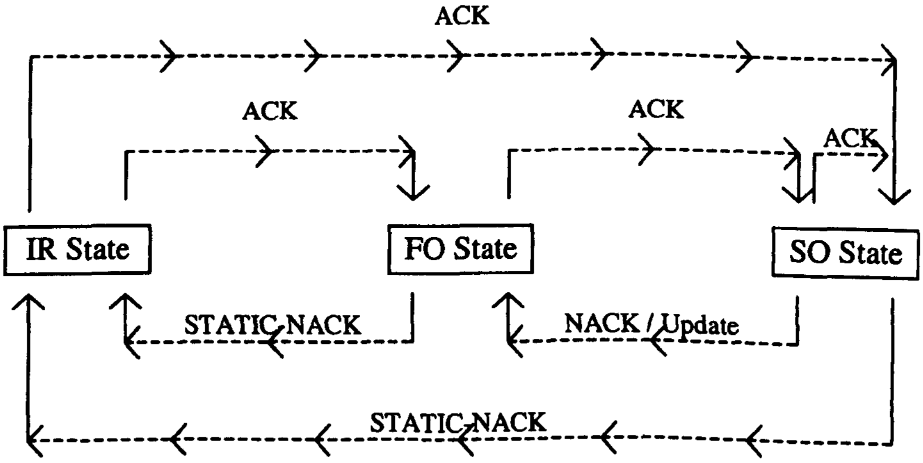


Figure 4-7: Bi-directional Reliable Mode and transition states

In R-mode, feedback is sent to acknowledge all context updates, including updates of the sequence number. This intensive usage of feedback channel gives R-mode maximum robustness against loss propagation and damage propagation. Although the R-mode minimises the probability of context invalidation, a larger number of damaged headers can be delivered when the context is invalidated. Figure 4-7 shows the details of the transitions between states and compression logic.

4.4 Feedback Channel

With ROCCO, the feedback channel is very important for achieving the maximum efficiency and robustness of the compression scheme. It is heavily reliant on the feedback channel. Without a feedback packet the performance of ROCCO is very poor in terms of protocol efficiency, and also packet loss rate performance.

The feedback channel carries information from de-compressor to the compressor. The following feedback messages are used.

- ***ACK*:** Acknowledges successful de-compression of a packet, which means that the context is up-to-date.
- ***NACK*:** Indicates that the dynamic context of the de-compressor is out of sync. Generated when several successive packets have failed to be decompressed correctly.
- ***STATIC-NACK*:** It is used when the static context of the de-compressor is not valid or has not been established.

4.4.1 Feedback Format

As mentioned before, feedback plays a very important role in the ROCCO scheme. It is one of the main factors in ensuring bandwidth efficiency and robustness. It can concatenate more than one feedback packet. Each feedback element has the following format, Figure 4-8.

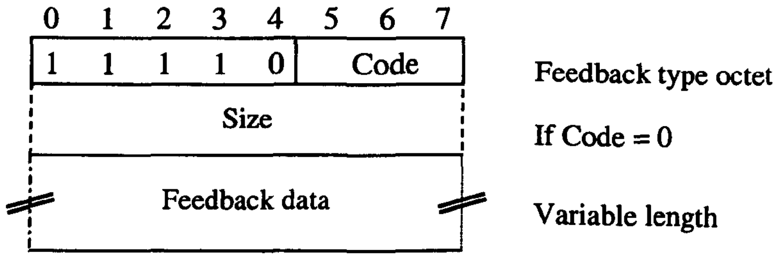


Figure 4-8: General FEEDBACK format structure

Code : 0 indicates that a Size octet is present.

1-7 indicates the size of the feedback data filed in octets.

Size : Optional octet indicating the size of the feedback data field in octets.

***Feedback data*:** Profile-specific feedback information. Includes CID information.

When the de-compressor has established the size of the feedback data field, the "feedback type octet" and the "size field" are not needed anymore. In that case, the format structure is changed as shown Figure 4-9.

FEEDBACK Data

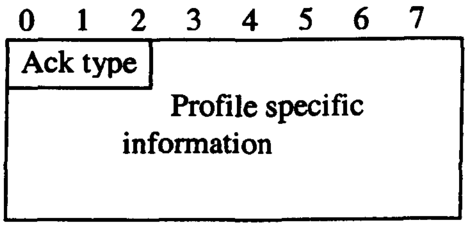


Figure 4-9: Changed FEEDBACK Data structure

When Ack type is:

00 = ACK

01 = NACK

10 = STATIC-NACK

11 = reserved

4.5 Encoding Techniques

There are significant differences between the encoding techniques used in ROCCO and CRTP. The CRTP algorithm uses a very simple delta encoding technique, whereas the encoding technique of ROCCO is more complicated. The encoding techniques are explained in detail in the following sections.

4.5.1 Least Significant Bits (LSB) Encoding

LSB encoding is used for header fields whose values are usually subject to change. With LSB encoding, instead of sending the original value, the *k* least significant bits of the field are transmitted. The de-compressor derives the original value by using the transmitted *k* bits in combination with a previously derived value as a reference, *v_ref*. The scheme is guaranteed to be correct if the compressor and the de-compressor each use the following interpretation intervals

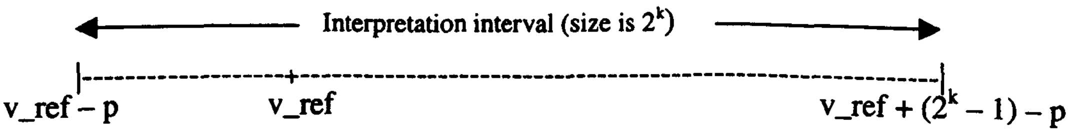
a) In which the original value resides,

b) In which the original value is the only value that has the exact same *k* least significant bits as those transmitted.

The Interpretation interval can be described as a function f(v_ref, k).

$$f\,(v\_ref, k) = [v\_ref - p,\ v\_ref + (2^k - 1) - p]$$

where, *p* is an integer.



The *k* least significant bits will uniquely identify a value in the f(v_ref, k) interval. The integer number, *p*, is used to shift the interpretation interval with respect to *v_ref*. The value of *p* therefore affects the encoding efficiency. It is chosen based on certain characteristics as listed below.

a) For field values that are always expected to increase,

$p = -1$,

the interpretation interval becomes

$$f\ (v\_ref, k) = [v\_ref + 1, v\_ref + 2^k]$$

b) For field values that are expected to stay the same or increase,

$p = 0$,

the interpretation interval becomes

$$f\ (v\_ref, k) = [v\_ref\ , v\_ref + 2^k - 1]$$

c) For field values that are normally constant, and are expected to increase only slightly,

$p = (2^{k-1}) - 1$,

the interpretation interval becomes

$$f\ (v\_ref, k) = [v\_ref - (2^{k-1}) + 1\ , v\_ref + 2^{k-1}]$$

d) For field values that feature small negative changes and a larger positive changes than the normal, such as the RTP timestamp (TS) for video or RTP sequence number (SN) when there is misordering,

$p = (2^{k-2}) - 1$,

the interpretation interval becomes

$$f\ (v\_ref, k) = [v\_ref - 2^{k-2} + 1, v\_ref + 3 \times 2^{k-2}]$$

The following section contains a simplified procedure for LSB compression and de-compression.

a) The compressor (de-compressor) always uses *v_ref_c (v_ref_d)*, the last compressed (de-compressed) value as reference, *v_ref.*

b) When compressing a value *v*, the compressor finds the minimum value of *k* such that "v" falls into the interval f (v_ref_c, k).

Where,

$k = g\ (v\_ref\_c, v)$ where, $g = \text{ceil}\ (\log_2|v\_ref - v|)$

The smallest *k* is picked that puts *v* in the interval f (v_ref_c, k).

c) When the de-compressor receives *m* LSBs, it uses the interpretation interval f(v_ref_d, m), which is called interval_d. It picks the decompressed value that in interval_d, whose LSBs match the received *m* bits.

The scheme suffers from very common mobile packet network problems, which are:

1. Packet loss between the compressor and the de-compressor
2. Transmission errors that are undetected by the lower layer.

These two factors drove the designers of the scheme to make it more complicated. In both cases, the synchronisation of *v-ref* between the compressor and de-compressor will be lost, and also the interpretation interval. Only if the value of *v* is still covered by the intersection *(interval_c, interval_d)*, will the de-compression be correct. In the case of undetected transmission errors, the corrupted LSBs give an incorrectly decompressed value, later used as *v_ref_d*, which is likely to lead to error propagation.

4.5.2 Scaled RTP Timestamp Encoding

The RTP TS (timestamp) does not increase randomly from packet to packet. Normally, the increment is an integer multiple of some unit, which is called TS_STRIDE here. This value is constant, and correlated with the sample rate of the application, as explained below.

In the case of "AUDIO"

The sample rate is normally 8 kHz and for AMR, one voice frame covers 20 ms. Assuming each frame is mapped one frame to one RTP packet, the RTP TS increment is always

→ $n \times (8000 \times 0.02) = n \times 160$, where n represents the packet number.

Therefore, the TS_STRIDE is 160. Note that the silence periods have no impact on this, as sample clock at the same source normally keeps running without changing either frame rate or frame boundaries.

In the case of "VIDEO"

The sample rate for most video codecs, such as MPEG-4, is 90 kHz [6]. With a fixed frame rate, of 30 frame/second, the RTP TS will increase by

→ $n \times (90000 / 30) = n \times 3000$ between video frames.

Therefore, the TS_STRIDE is 3000. Video frame sizes are often large. Therefore, they may be divided into several RTP packets. This increases the robustness against the packet loss. In this case all RTP packets that belong to the same video frame, carry the same RTP TS.

When scaled RTP Timestamp encoding is used, the TS is downscaled by a factor of TS_STRIDE before compression. As a result of this, for each compressed TS, Floor ($\log_2$ (TS_STRIDE) bits are saved.

The number of saved bits = Floor ($\log_2$ (TS_STRIDE) bits Equation 4.1

At the session start, the compressor sends the value of TS_STRIDE to the de-compressor, and the absolute value of several TS fields. This way, the de-compressor initialises its database and calculates the TS_OFFSET by using the expression below.

TS_OFFSET = (absolute TS value) module TS_STRIDE Equation 4.2

After the initialization procedure, when the compressor can be confident that the de-compressor has received the information, the compressor stops operation on the original TS. Instead, it compresses the downscaled value of TS, which is calculated by using the Equation 4.3. LSB encoding is always used to compress these values.

TS_SCALED = TS / TS_STRIDE Equation 4.3

When the de-compressor receives the compressed value of TS_SCALED, it first derives the original TS_SCALED value. Then, using the Equation 4.4, the de-compressor calculates the original RTP TS:

TS = TS_SCALED × TS_STRIDE + TS_OFFSET Equation 4.4

4.6 Contexts and context Identifiers

Contexts are identified by a context identifier, CID, which is sent along with the compressed headers and feedback information. Context information is kept in a context table, which is indexed using the CID, as in the CRTP header compression scheme. However in ROCCO, each CID number represents specific context information for a particular channel. With ROCCO, different channels can have the same CID numbers, but the CID numbers do not necessarily refer to the same context. This makes a CID number unique within a channel. The CID can take values between 0 and $2^{12}- 1 = 4095$. This CID space is negotiated between compressor and the de-compressor when a channel is established.

4.7 Packet Formats

The ROCCO scheme has three link modes, which are U-mode, O-mode, and R-mode. These are explained in Section 4.3. This section covers the packet formats that are used within these modes. The initial bits within the first byte of each packet always used to identify the packet formats as shown below. It should be noted that, the RFC's ROCCO does not include the *reference_header packet format.* This is a new packet format proposed in this thesis.

11110	: feedback
11111000	: IR-DYN packet
1111110	: IR packet
11111111	***: Reference_header (proposed by the author)***
Other	: compressed header

4.7.1 IR Packet format:

This packet type communicates the static part of the context, although it can optionally also communicate the dynamic part of the context. Its format is:

1	1	1	1	1	1	0	X	
Co	*Ci*	*Ro*	*Ri*	4-bits CID				
CID								
Profile								
CRCs								
Static Fields								
CRCd								
Dynamic Fields								If X = 1

Figure 4-10: IR Packet Format

As shown in Figure 4-10, the first seven bits which are "1111110", are used to identify the packet format. The last bit of the first byte is used to flag whether the packet is carrying dynamic fields' information or not. The first two bits of the second byte, *Co* and *Ci* are used to identify how many bits of CID are carried with the packet, (see Table 4-1). The 3rd and 4th bits, *Ro* and *Ri*, are used to confirm whether the ARUS technique, which is proposed by author in section 4.8, is active within the packet stream (see Table 4-2). The last 4 bits are allocated for the CID. The third byte may be allocated for the CID number. It is only included when the CID number is long, as indicated by *Co* and *Ci*. *(It increases the CID number from 4 to 12 bits.)* The CID number is followed by the Profile number, the static fields, and then the dynamic fields of the protocols in their original order. (i.e. RTP-UDP-IP). Two 8-bit CRCs are used. The first one, CRCs, is allocated just before the static fields, which is calculated based on the full static headers. The second one, CRC, is used when the header carries dynamic fields. It is placed just after the static fields and is calculated on the dynamic fields of the original headers. The final part of the packet contains the payload.

Co	***Ci***	***Number of CID its***
0	0	4-bits CID
1	0	Reserved
0	1	Reserved
1	1	12-bits CID

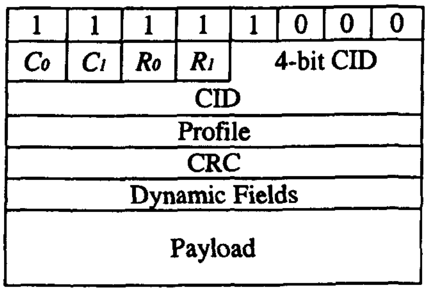
Table 4-1: CID bits code

Ro	***Ri***	***ARUS***
0	0	Inactive
1	0	Reserved
0	1	Reserved
1	1	Active

Table 4-2: Reference bits code

4.7.2 IR-DYN header:

The IR-DYN header is similar to the IR packet. The only difference between the two is, that the IR_DYN packets do not carry the static fields of the protocols. It only carries the dynamic fields. The 8-bit CRC is calculated on the original uncompressed full header. This header is always used after the static fields have been initialised. Any unexpected changes in the headers are transmitted with this packet. It is often used to update the dynamic fields at the de-compressor. The Figure 4-11 shows the IR-DYN header format.

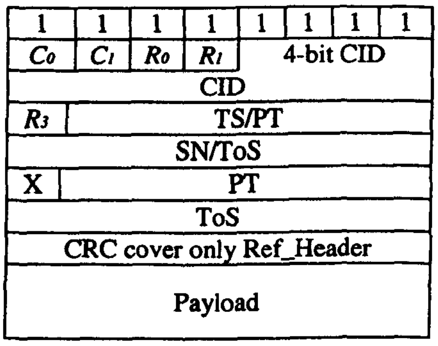


1	1	1	1	1	0	0	0
C_0	C_1	R_0	R_1	4-bit CID			
CID							
Profile							
CRC							
Dynamic Fields							
Payload							

Figure 4-11: IR-DYN header format

4.7.3 Adaptive_Reference Header:

The Adaptive_Reference Header is a new header format that is proposed in this work. Figure 4-12 shows the Adaptive_reference header format. It is used to adaptively confirm the change or update the fields that are shown in Table 4-3. The compressor observes which fields change more frequently and transmit these fields more often than the others. However, all fields are transmitted at least once every 50 packets. In section 4.8, this is explained in more details



1	1	1	1	1	1	1	1
C_0	C_1	R_0	R_1	4-bit CID			
CID							
R_3	TS/PT						
SN/ToS							
X	PT						
ToS							
CRC cover only Ref_Header							
Payload							

Figure 4-12: Adaptive_reference header format

R_0	***R_1***	***R_3***	
1	0	0	TS + SN
1	0	1	PT + SN
0	1	0	PT + ToS
0	1	1	SN + PT
1	1	0	TS + SN + ToS
1	1	1	TS + SN + PT + ToS

Table 4-3: Reference field identifier bits

4.7.4 General Compressed Formats

The general compressed format is shown in Figure 4-13. Unlike IR and IR-DYN packets' format, the format varies according to the state of compression and mode of transition. Mainly, it is the first and second (or third, if the large CID is used) octets that change. The main concern is to optimise the overhead according to each transmission mode. The first 4 bits in the second octet, *C_0*, *C_1*, *R_0* and *R_1* are proposed by the author. The IETF standard ROCCO [6] does not have this field. There are three types of compressed header, which are "type 0", "type 1", and "type 2". Each type has a different format for each mode. The type of the packet is always represented by the first two bits of the first octet. In subsequent sections, these formats are presented. The modes are:

U – Unidirectional mode

O- Bidirectional Optimistic mode

R- Bidirectional Reliable mode

<table>
<tr><td colspan="5">1st Octet of Base Header</td></tr>
<tr><td>C_0</td><td>C_1</td><td>R_0</td><td>R_1</td><td>4-bits CID</td></tr>
<tr><td colspan="5">CID</td></tr>
<tr><td colspan="5">Remaining Octets of base header</td></tr>
<tr><td colspan="5">Payload</td></tr>
</table>

Figure 4-13: General Compressed Format

4.7.4.1 Packet Type "0":

R-0: minimum 1 octet

0	1	2	3	4	5	6	7	
0	0	SN					Xc	When Xc = 1, CID present
C_0	C_1	R_0	R_1	4-bits CID				When $C_0 = C_1 = 0$, 4 bits CID
CID								if $C_0 = C_1 = 1$, 12-bits CID
Reference SN								When $R_0 = 1$
Reference TS								When R1 = 1

Figure 4-14: Packet type "0" format for Reliable Mode

1	CRC

Figure 4-15: Minimum Overhead format

This format is used for bi-directional reliable mode, and is represented by "00" in first two bits for this format. The last bit in the first octet, X_c, is used to indicate whether the CID is included or not. The first two bits of the second octets, C_0 and C_1, represent the size of the CID, which is used. The 3rd and 4th bits, R_0 and R_1, are used for referenced compressed scheme. The R_0 is set to one when a reference SN is transmitted, and R_1 is set to one when a reference TS is transmitted. If both of them are set one, this means both reference SN and TS are transmitted. Both fields are compressed based on the last reference field. Feedback is compulsory when headers carry any reference field.

R-0-CRC: minimum 2 octets

0	1	2	3	4	5	6	7	
0	1	SN					Xc	When Xc = 1, CID present
C_0	C_1	R_0	R_1	4-bits CID				When $C_0 = C_1 = 0$, 4 bits CID
CID								if $C_0 = C_1 = 1$, 12-bits CID
CRC								8-bit CRC
Reference SN								
Reference TS								

Figure 4-16: Packet type "0" with CRC format for Reliable Mode

R-O-CRC is same as “R-0”, except for the 8-bit CRC that is calculated over the original uncompressed headers. For this format, the first two bits are set to “0 1”. The rest operate exactly as “R-O”

UO-0: 1 octet

0	SN	CRC

3-bit CRC

Figure 4-17: Packet type “0” format for Unidirectional and Optimistic Mode

This is used in Unidirectional and Bi-directional Optimistic mode. It is the minimum size header. The first bit is set to “0” for this format, and a 3-bit CRC is used. This packet format is used when compression and de-compression is based on a uniform pattern, and header is completely predictable for a given SN (RTP sequence number). It can be independently compressed and de-compressed on the basis of previous context information.

4.7.4.2 Packet Type “1”;

R-1: minimum 3 octets

0	1	2	3	4	5	6	7
1	0	SN					Xc
C_0	C_1	R_0	R_1	4-bits CID			
CID							
M	Xr	TS					
Reference TS							
Reference SN							

When Xc = 1, CID present
When $C_0 = C_1 = 0$, 4 bits CID
if $C_0 = C_1 = 1$, 12-bits CID
Xr = reserved, M = Marker

Figure 4-18: Packet type “1” format for Reliable Mode

This configuration is used in reliable mode, when the packet type is 1. It is indicated by the first two bits within the first octet. These bits are “1 0” for this type. It is the same as R-0, without the CRC. Instead, the RTP marker bit and compressed TS are included after the CID field.

U/O-1: min 3 octets

0	1	2	3	4	5	6	7	
1	0	TS						
C_0	C_1	R_0	R_1	4-bits CID				When $C_0 = C_1 = 0$, 4 bits CID
CID								if $C_0 = C_1 = 1$, 12-bits CID
M	SN				CRC			3-bit CRC
Reference TS								When $R_0 = 1$, when ref_header is used
Reference SN								When $R_1 = 1$, when ref_header is used
Xr	Reference CRC							If $R_0 = R_1 = 1$, 7-bit CRC, Xr = reserved

Figure 4-19: Packet type "1" format for Unidirectional and Optimistic Mode

U/O-1 is used in unidirectional and bi-directional optimistic mode. Its size is a minimum of three octets. It is represented by "1 0". This header always carries the CID number and a 3-bit CRC. The second 7-bit CRC is included when both R_0 and R_1 are set to 1. It is calculated over the original reference header.

4.7.4.3 Packet Type "2";

For all Modes UOR-2; at least 4 octets

0	1	2	3	4	5	6	7	
1	1	1	SN					
C_0	C_1	R_0	R_1	4-bits CID				When $C_0 = C_1 = 0$, 4 bits CID
CID								if $C_0 = C_1 = 1$, 12-bits CID
M	Xr	TS						Xr = reserved
CRC								8-bit CRC
Reference TS								When $R_0 = 1$, when ref_header is used
Reference SN								When $R_1 = 1$, when ref_header is used
Xr	Reference CRC							If $R_0 = R_1 =1$, 7-bit CRC, Xr = reserved

Figure 4-20: Packet type "2" format for All Mode

This header is a common header type for all modes, see Figure 4-20. It is indicated by setting the first three bits if the header to 1. The last five bits in the first octet are used for the RTP sequence number. The second and optional 3rd octets are the same as for the other compressed header formats. After the CID number, the next octet starts with the RTP marker bit, and the second bit is reserved for future use. The last 6 bits of this octet contain the compressed TS field. This is followed by 8-bit CRC, which is calculated over the original uncompressed header. The last three octets are new proposed optional headers and they are included, as for the U/O-1 type.

4.8 Adaptive_reference update scheme (ARUS)

The *Adaptive_Reference Update Scheme* (ARUS) is proposed by the author to minimise/stop the risk of context damage propagation within the ROCCO algorithm. As explained at the beginning of the chapter, the LSB encoding is based on the last original value at the compressor and the last reconstructed value at the de-compressor. It is clear that correct reconstruction of all headers depend on the previously reconstructed header. The ROCCO includes CRC bits in all headers, which are calculated on the original transmitted uncompressed headers, to minimise the risk. However, if undetected errors propagate, all of the context will be updated incorrectly. In addition, any error on the CRC-bits will cause packet to be discarded by the de-compressor. In real-time applications, (i.e., interactive or streaming), this is not affordable.

In the ARUS scheme, the compressor identifies the reference header based on the fields' irregular changes or timeout. Normally, once the call is established and the compressor and the de-compressor initialise their context, the changes in the header fields are known and expected. Sometimes due to packets lost on the uplink, or the late arrival of a packet due to core network congestion (explained in detail in chapter 5), there are unexpected changes between consecutive packets at the compressor on the downlink. With standard ROCCO compression, these headers are encoded using the last transmitted header as a reference. The ARUS scheme on the other hand, picks a header with irregular changes and treats it as a reference header. The space between two reference_headers can vary depending on how often the unexpected changes occurs, and also the channel conditions. The adaptively chosen reference header is transmitted by taking the difference between two consecutive reference headers and using LSB encoding to transmit the differences. The de-compressor always sends an ACK when it receives a reference header. The performance is presented and discussed in section 4-9. Note that, these reference headers are treated as high priority packets and use a Prioritisation scheme to transmit them, which is explained in next section.

4.9 Prioritisation

One technique capable of delivering good error resilience performance for a variety of applications is Unequal Error Protection (UEP) [32] [38]. These schemes [32] [38] protect fixed lengths of data with different strength channel codes, with data at the start of the packet receiving the greatest protection. However, when full headers are sent, the amount of important header data at the beginning of the packet grows in size. This results in some of the header information receiving less protection than required. A similar method for improving error resilience is the prioritisation of different packets. The network sends the packets using channels with different priorities, allocating more important packets to more reliable channels, because the packets have different levels of sensitivity to channel errors. This scheme increases robustness, but may cost more in terms of bandwidth requirements.

Header compression divides the header into three formats, which are full header, refreshed header and compressed header. It is a simple task to produce two priority classes from this arrangement, using the full header packets and reference header packets as the high priority class and the compressed header packet as a lower priority class. This arrangement of packet information into different classes will allow Unequal Protection mechanisms to be employed for wireless services in a UMTS (Universal Mobile Telecommunications System) scenario. The 3rd Generation mobile access technologies will offer a number of bearer services to the user. These will be defined in terms of the bearer characteristics and bearer quality. The latter includes parameters such as end-to-end latency, delay variation, throughput and bit and frame error rates.

Bearer channels do not offer any particular type of service, such as speech, data transfer protocols or video services, but instead offer links conforming to a given set of Quality of Service parameters. It is required that a single terminal should have access to more than one bearer channel, with each separate channel being able to offer different levels of Quality of Service.

In the scheme being proposed, each packet class will be transported over a different mobile bearer channel as offered by the underlying network. As long as the bearer channels can meet the QoS parameters set by the application, then the prioritisation scheme is completely independent of the underlying network. This facilitates interoperability between similarly configured terminals communicating via different networks.

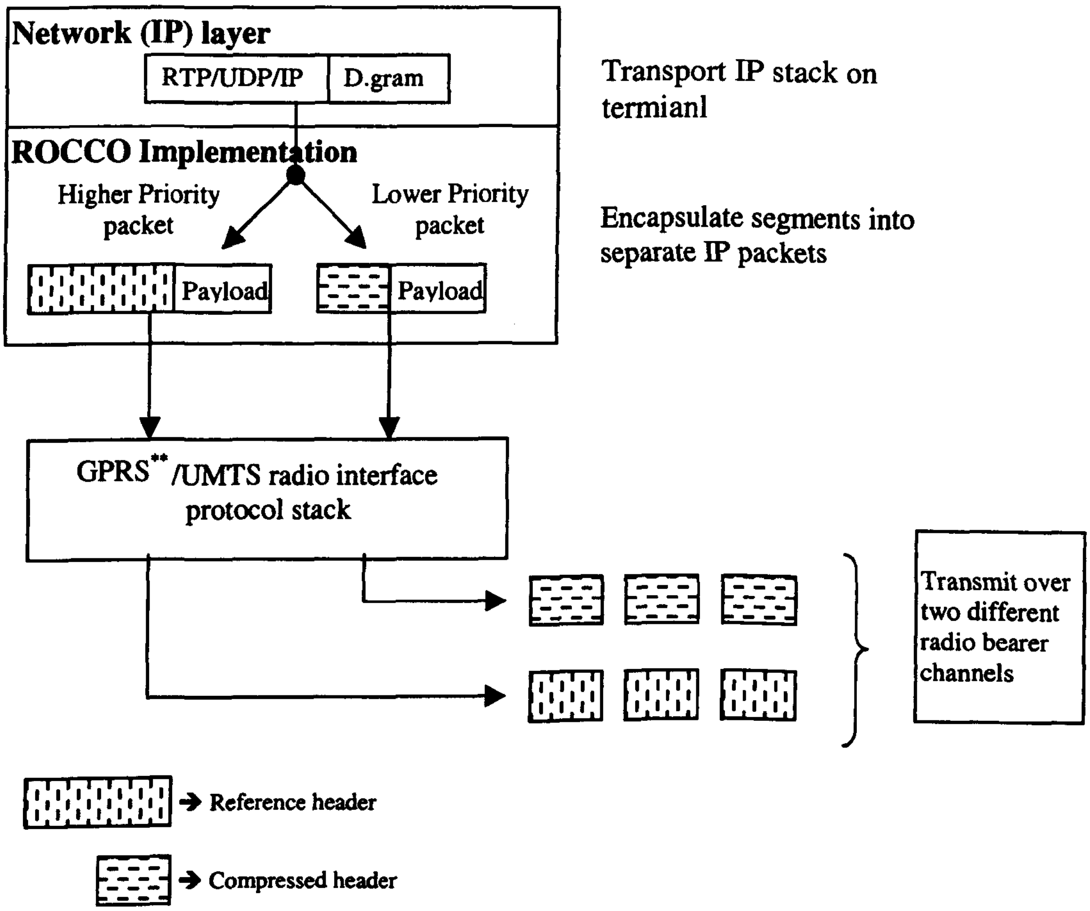

Network (IP) layer
RTP/UDP/IP
D.gram
Transport IP stack on termianl
ROCCO Implementation
Higher Priority packet
Lower Priority packet
Payload
Payload
Encapsulate segments into separate IP packets
GPRS** /UMTS radio interface protocol stack
Transmit over two different radio bearer channels
→ Reference header
→ Compressed header

Figure 4-21: Prioritisation scheme

QoS management will be carried out by integrating the end-to-end requirements set by the end applications with the mechanisms employed by the underlying networks. In this scenario, UEP is implemented by transmitting the two different packets over different bearer channels meeting different QoS levels.

** Note: Only Release 99 and later GPRS supports multiple PDP contests

4.10 Simulations and Analysis of the Results

In this section, the simulation results for ROCCO and the proposed ARUS enhancements are given along with discussions. The simulation conditions are described within the section. The results are presented in a comparative way, which clearly identifies the performance improvement possible with the two types of resilience schemes. The main performance metrics, packet loss rate and protocol efficiency, are examined in this section. In addition, the differences between CRTP and the ROCCO compression scheme are presented. The effect of the header compression schemes upon speech and video quality over wireless link are also presented. Note that the following results focused on the compression efficiency and robustness. The effects of Internet congestion and consequent late arrival of packets are not examined. It is assumed that all packets arrive on time. A packet with errors in the payload is not regarded as lost, unless the compressed header gets corrupted and cannot be used to reconstruct the original header.

The performance of the header compressions mechanism in cellular environments was evaluated using the GPRS access network as a case study. The experiments described in this section employ the CS-2 code at a carrier frequency of 1800MHz, using the TU50 (Typical Urban Scenario, mobile terminal velocity of 50kph) Multipath model as specified in experiments.

C/I (dB)	BER
7	1.6×10^{-2}
9	3.7×10^{-3}
12	2.4×10^{-5}
15	1.0×10^{-6}

Table 4-4: BER against C/I with CS-1 code

C/I (dB)	BER
7	7.30×10^{-2}
9	2.96×10^{-2}
12	4.90×10^{-3}
15	5.43×10^{-4}

Table 4-5: BER against C/I with CS-2 code

Table 4-4 and 4-5 shows the average BERs of the CS-1 and CS-2 code respectively with different Carrier-to-Interference Ratios (dB).

4.10.1 Experiments and Results with standard ROCCO

The first simulations were carried out using 9000 speech packets to investigate the performance of ROCCO in terms of packet loss rate and protocol efficiency. The packets are generated with a fixed payload size, 160 bits, which represents 20 ms speech frame length with an 8 kb/s bit-rate. The frames are sent with one-frame-to-one RTP packet mapping.

As the bit error rate increases on the channel, the average overhead proportionally increases with both standard header compression schemes. The consequence of this is an increase in packet size due to the irregularly varying overhead. Effectively, the required bandwidth (BW) is increased over noisy channels.

4.10.1.1 Packet Loss Rate

It has been mentioned that compression schemes increase the error susceptibility of data because compression removes the redundancy from consecutive packets. Thus, when header compression is employed for joint RTP/UDP/IP headers, error susceptibility is increased as well. Figure 4-22 demonstrates the packet loss rate differences between standard ROCCO, standard CRTP, and improved CRTP using reference header and the no-header compression applications. The results are shown against the C/I ratio of the simulation channel. The packet is regarded as being 'lost' if it is not passed up to the application layer (speech/video codec), which means that as long as the de-compressor succeeds in reconstructing the headers, the packet is considered to be intact, even if there is an error in the payload.

The results show that there is a significant packet loss rate improvement with ROCCO even compared with the best CRTP scheme (using reference-header), which is proposed by the author in chapter 3. The IETF standard CRTP [1] produces higher packet dropping rates at all C/I ratios compared to the other IETF standard ROCCO [6]. For example, the average packet loss rate with CRTP at 7 dB C/I is obtained when the Reference_header packet is employed. The loss rate is about 25%, whereas with ROCCO, the packet loss rate is 8.72%. Therefore, ROCCO is more robust than CRTP. It is clear that, without any compression the packet loss rate results are the worst even though there is no dependency between packets. This is because the RTP/UDP/IPv4 headers are very susceptible to channel errors. Their size makes error occurrences very likely, thereby causing a packet loss.

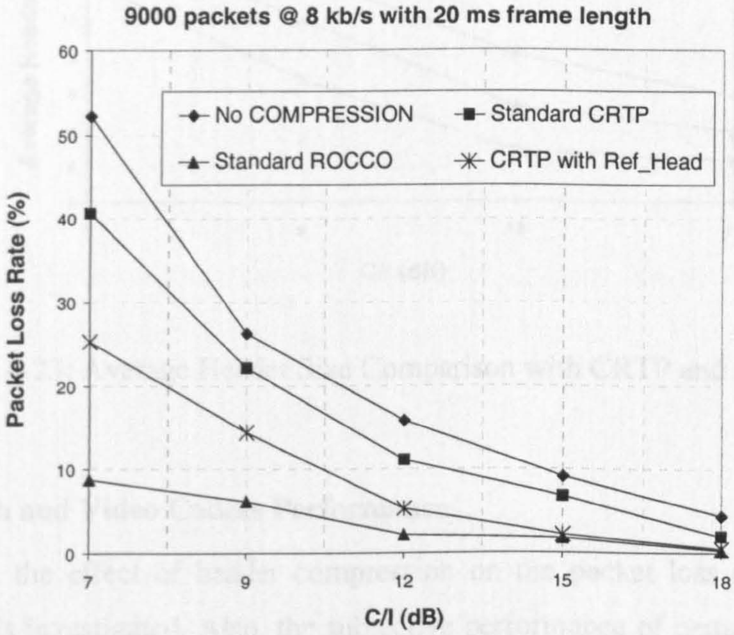


Figure 4-22: Packet Loss Rate Performance with ROCCO and CRTP

4.10.1.2 Protocol Efficiency

The required header size is critical for mobile channels. Figure 4-23 shows the average header size per packet versus C/I ratio. The results show that with ROCCO, the required average header size is smaller than with CRTP. For example, at 7 dB C/I, the CRTP compression scheme with compressed reference_header reduces the average header size from 40 bytes (full header) to approximetry 15 bytes per packet, which is 42 % of one packet. However, with ROCCO, the header is reduced to about 12 bytes per packet, which is about 37.5 % of the packet size. For any channel condition, the required overhead is reduced significantly. Although ROCO can theoretically reduce the compressed header size to one byte, even with 15 dB C/I ratio (5.43×10^{-4} bit error rate), the average header size is more than two bytes. The main reason for this is the time taken for the transition between the mode and state of compression.

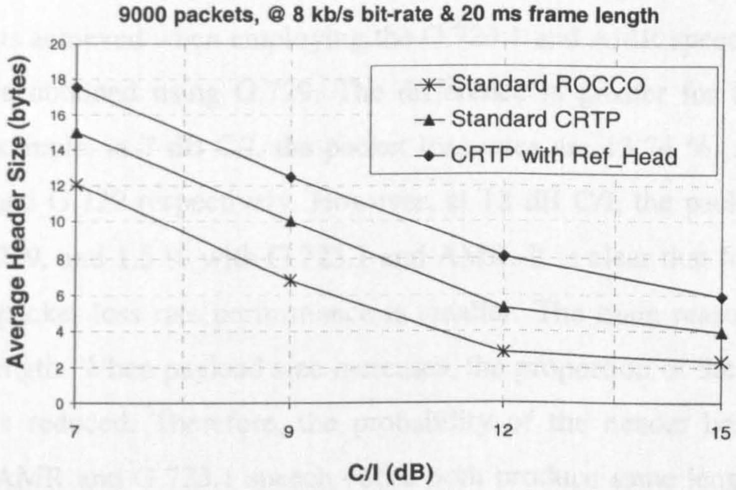


Figure 4-23: Average Header Size Comparison with CRTP and ROCCO

4.10.1.3 Speech and Video Codecs Performance

In this section, the effect of header compression on the packet loss rate for different speech codecs is investigated. Also, the subjective performance of certain speech codecs and a video codec is evaluated for transmission over GPRS channels using the CS-1 coding scheme.

The simulations were carried out using three different ITU-T standard speech codecs and a fully resilient video codec developed within the group. The speech codecs are: G.723.1 with 30 ms frame length; AMR with 20 ms frame length; and G.729 with 10 ms frame length. Their bit-rates are 5.3 kb/s, 7.95 kb/s, and 8 kb/s for G.723.1, AMR, and G.729 respectively. The MPEG-4 video codec uses a bit rate of 48kbits/s and a frame rate of 25 frame/s.

As shown in Figure 4-24, the use of different speech codecs with different characteristics, affects the packet loss rate performance. The results are obtained over 1600 packets for each speech codec. The speech frames are sent with one-frame-to-one RTP packet mapping. Because of the different bit-rates and frame length of these speech codecs, 1600 packets correspond to a different time period for each. 1600 speech packets correspond to 16 sec, 32 sec, and 48 sec speech with G.723.1, AMR, and G.729 respectively.

Figure 4-24 demonstrates the packet loss rate using different speech codecs. The lowest packet loss rate is achieved when employing the G.723.1 and AMR speech codec, and the worst results are obtained using G.729. The difference is greater for higher error rate channels. For example, at 7 dB C/I, the packet loss rates are 13.74 %, and 6.17 % with G.723.1-AMR and G.729 respectively. However, at 12 dB C/I, the packet loss rates are 2.92 % with G.729, and 1.5 % with G.723.1 and AMR. It is clear that for better channel conditions, the packet loss rate performance is smaller. The main reason for this is the relative frame length. When payload size increases, the proportion of the packet taken up by the header is reduced. Therefore, the probability of the header being corrupted is reduced. Since AMR and G.723.1 speech codes both produce same length of frame, the effect of the packet loss rate due to the header corruption under same circumstances is same.

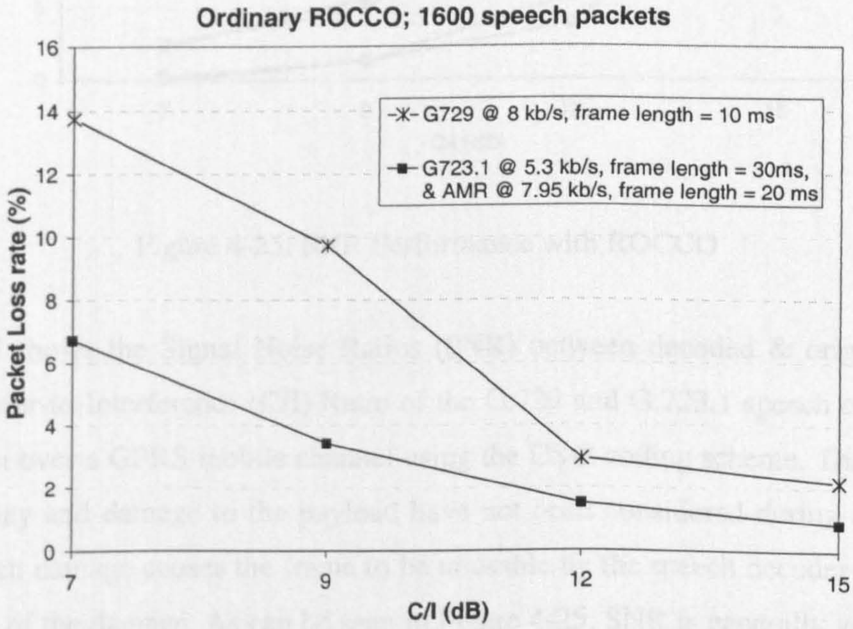


Figure 4-24: Packet Loss Rate Performance with various speech codecs

Speech Quality:

In Figure 4-25, the speech codec quality is plotted using different header compression. In the precious section, it was shown that, a speech codec with a shorter speech frame incurs a higher packet loss rate under the same channel conditions and header compression scheme. However, Figure 4-25 demonstrates this does not necessarily translate into that better speech quality. For example, G.729 incurs higher packet loss rate but is more robust against packet loss compared to G.723.1.

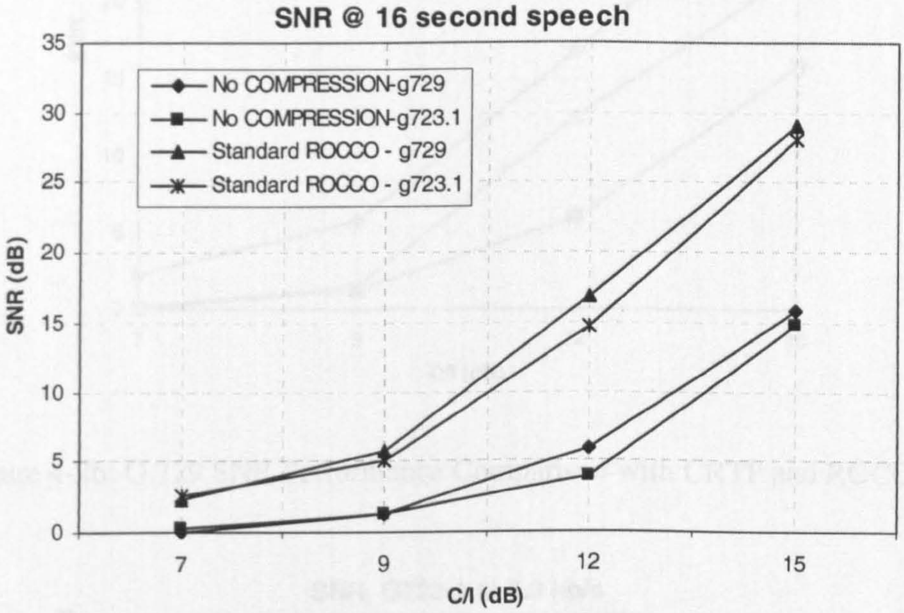


Figure 4-25: SNR Performance with ROCCO

Figure 4-25 shows the Signal Noise Ratios (SNR) between decoded & original speech against Carrier-to-Interference (C/I) Ratio of the G.729 and G.723.1 speech codecs, after transmission over a GPRS mobile channel using the CS-2 coding scheme. The round-trip (packet) delay and damage to the payload have not been considered during experiment. Whether such damage causes the frame to be unusable by the speech decoder depends on the location of the damage. As can be seen in Figure 4-25, SNR is generally very low due to packet loss, particularly when C/I<12dB. The results show that standard ROCCO performs much better than standard CRTP. The quality results show that the performance of standard ROCCO is very close to that of the enhanced CRTP scheme, see Figure 4-22. The intelligibility of the speech is much better with ROCCO according to informal subjective testing. This means that, the choice of header compression scheme is as important as the speech codec choice in VoIP for wireless packet communications.

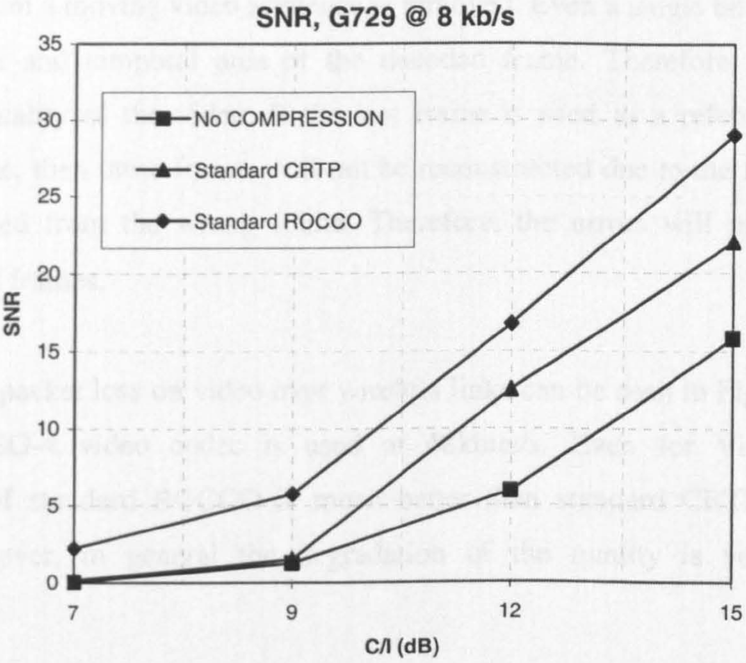


Figure 4-26: G.729 SNR Performance Comparison with CRTP and ROCCO

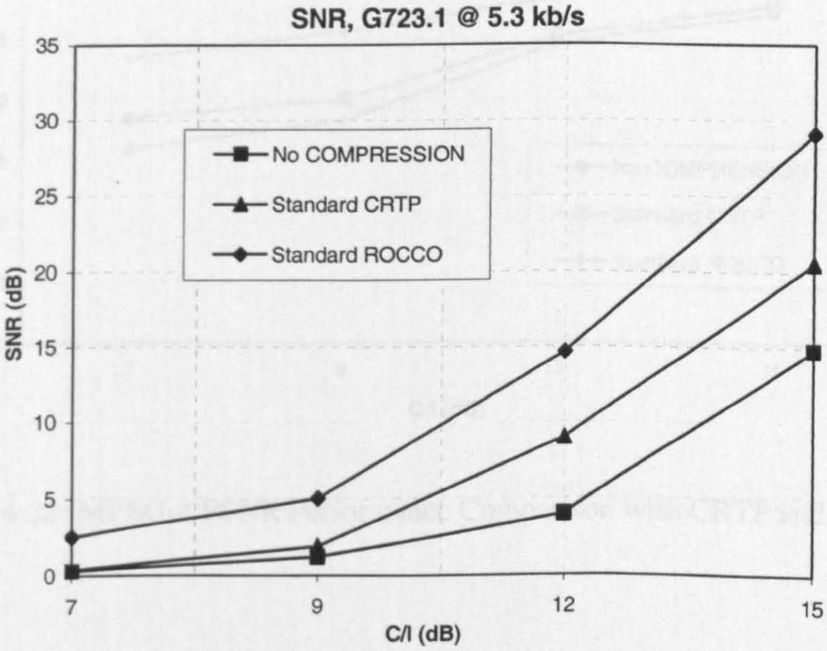


Figure 4-27: G.723.1 SNR Performance Comparison with CRTP and ROCCO

Video Quality:

Even a relatively low level of packet losses can have a severe effect on the quality of decoded video data. The reason for this is that much of the inherent special and temporal redundancy from a moving video sequence is removed. Even a single bit error can corrupt a large spatial and temporal area of the decoded frame. Therefore, packet loss will degrade the quality of the video. If the lost frame is used as a reference for a future predicted frame, then these frames will not be reconstructed due to the lost frame or will be reconstructed from the wrong frame. Therefore, the errors will propagate through future decoded frames.

The effects of packet loss on video over wireless links can be seen in Figure 4-28. In this test, the MPEG-4 video codec is used at 48kbits/s. Even for Video-over-IP, the performance of standard ROCCO is much better than standard CRTP over a mobile channel. However, in general the degradation of the quality is very obvious and objectionable.

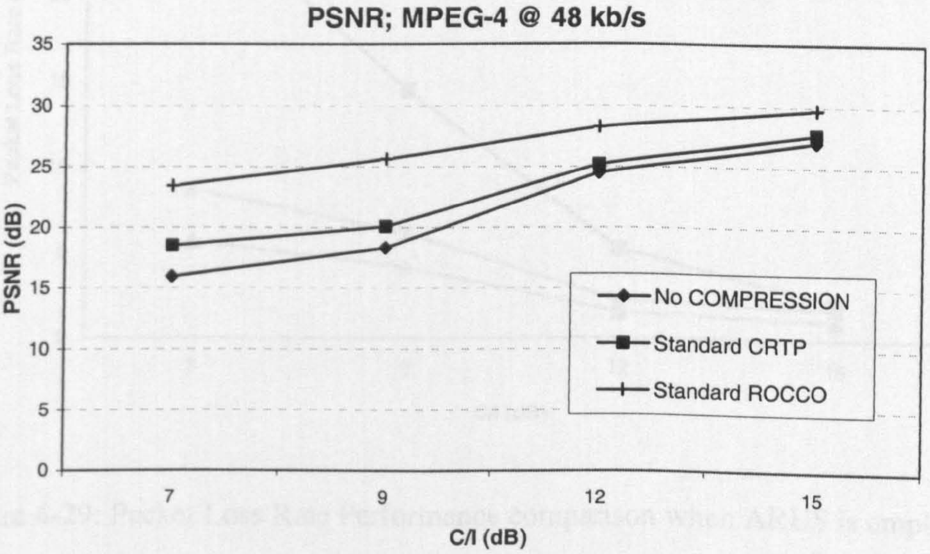


Figure 4-28: MPEG-4 PSNR Performance Comparison with CRTP and ROCCO

4.10.2 Experiment and Results with ARUS

In this section, the effects of the proposed ARUS scheme on the packet loss rate and quality performance for speech/video applications was investigated. The simulation scenario is the same as the previous section.

4.10.2.1 Packet Loss Rate

Figure 4-29 shows that there are significant improvements in packet loss rate, when the Adaptive_reference scheme is employed with ROCCO. As can be seen in Figure 4-29, the effect of the ARUS is significant particularly over higher bit error rate channels. However, this performance difference gets smaller as the channel condition improves.

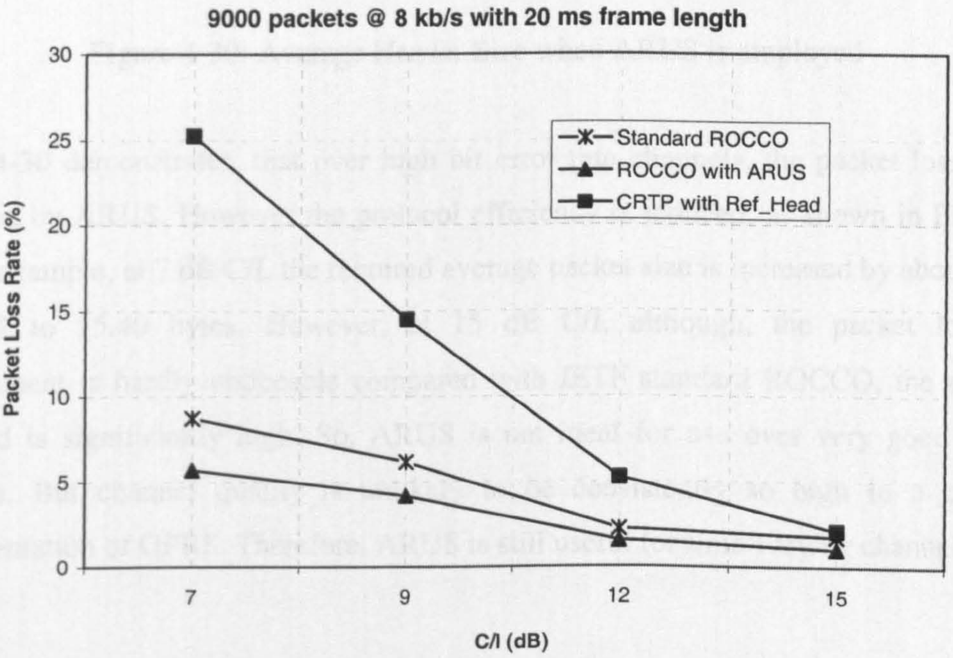


Figure 4-29: Packet Loss Rate Performance comparison when ARUS is employed

4.10.2.2 Protocol Efficiency

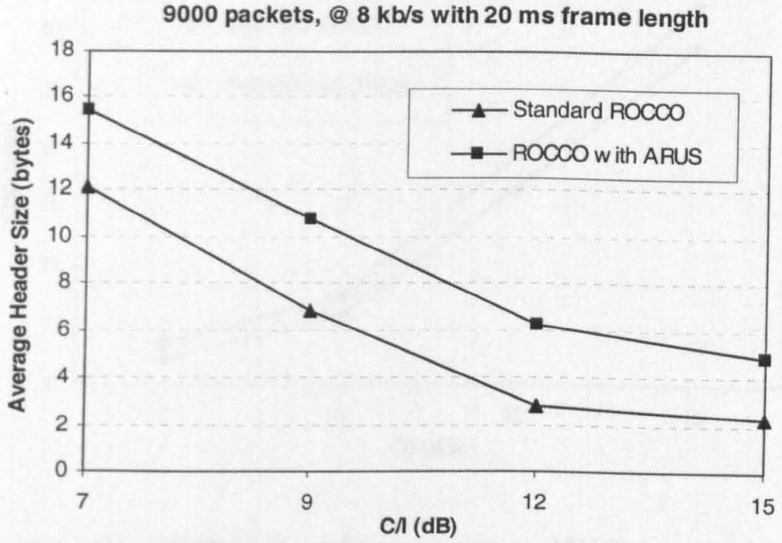


Figure 4-30: Average Header Size when ARUS is employed

Figure 4-30 demonstrates, that over high bit error rate channels, the packet loss rate is improved by ARUS. However the protocol efficiency is reduced, as shown in Figure 4-30. For example, at 7 dB C/I, the required average packet size is increased by about 30 %, from 12 to 15.40 bytes. However, at 15 dB C/I, although, the packet loss rate improvement is hardly noticeable compared with IETF standard ROCCO, the required overhead is significantly high. So, ARUS is not ideal for use over very good quality channels. But channel quality is unlikely to be consistently so high in a practical implementation of GPRS. Therefore, ARUS is still useful for time-varying channels.

4.10.2.3 Speech and Video Codecs Performance

In this section, the impact of ARUS on speech and video quality is investigated and the quality performances are plotted against the C/I ratio of a GPRS channel using the CS-2 coding scheme. Figures 4-31 and 4-32 are showing the SNR results for G.729 and G.723.1 respectively. In both cases, the ARUS improve the SNR quality all the time. With G.729, SNR improves by 1.18 dB with a C/I of 9 dB. With G.723.1, there is about 2 dB improvements with a C/I of 9dB.

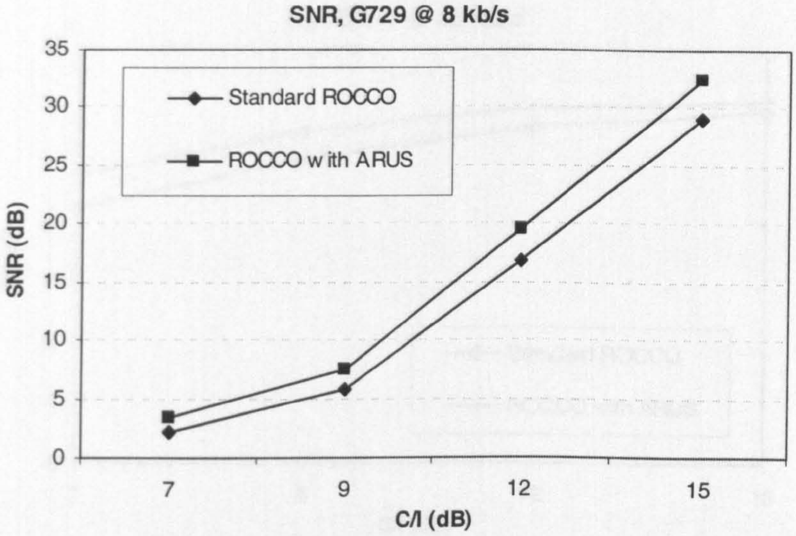


Figure 4-31: G729 SNR performance when ARUS is employed

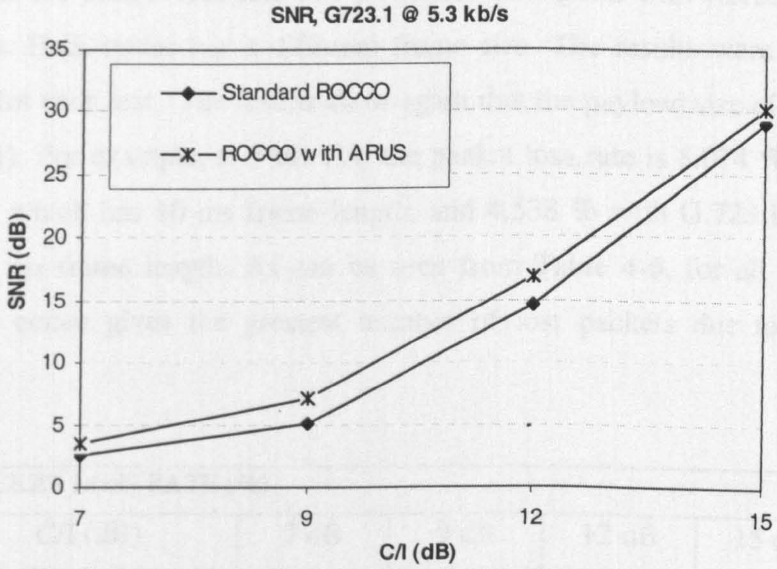


Figure 4-32: G.723.1 SNR performance when ARUS is employed

Figure 4-33 shows that when using ARUS, the video quality improves, especially in high bit error rate channels. When the channel quality improves, the improvement in video quality is reduced. At 7 dB C/I, the improvement is about 2.5 dB, whereas it is about 0.78 dB with C/I of 15dB.

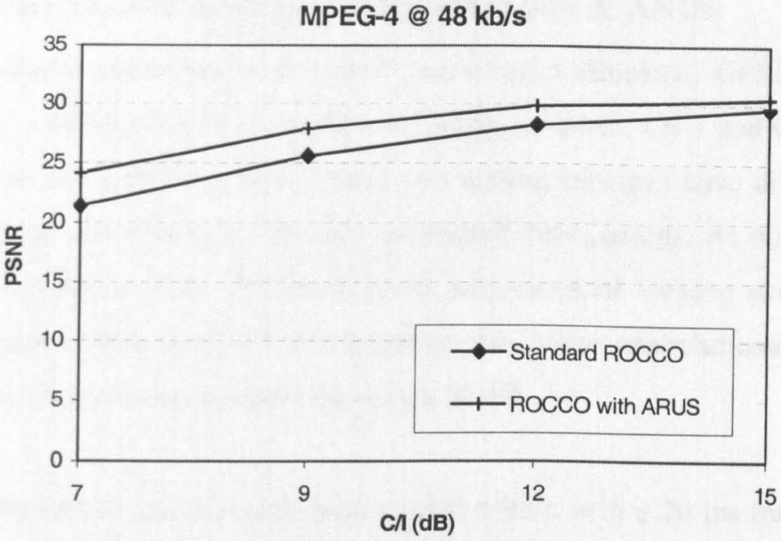


Figure 4-33: MPEG-4 PSNR performance with ARUS

Table 4-6, lists the packet loss rate due to header corruption with ARUS using various speech codecs. Each codec has a different frame size. The results were obtained using 1600 packets for each test. The results show again that the payload size affects the packet loss rate (PLR). For example, at 7 dB C/I, the packet loss rate is 8.074 %, with a G.729 speech codec, which has 10 ms frame length, and 4.538 % with G.723.1 speech codec, which has 10 ms frame length. As can be seen from Table 4-6, for all C/Is, using the G.729 speech codec gives the greatest number of lost packets due to the corrupted overhead.

PACKET LOSS RATE (%)				
C/I (dB)	7 dB	9 dB	12 dB	15 dB
No COMPRESSION	52.183 %	26.306 %	15.860 %	9.175 %
ROCCO with ARUS				
G.729	8.074 %	5.633 %	2.186 %	1.250 %
G.723.1	4.538 %	2.273 %	1.241 %	0.799 %

Table 4-6: Packet Loss Rate performance with ARUS for various codecs

4.10.3 Experiments and Results with Prioritisation & ARUS

The prioritisation experiments ware carried out using a simulated GPRS mobile access channel with a combination of two different coding schemes, CS-1 and CS-2, within the same packet stream communication. These two coding schemes have different levels of protection against transmission error the transmitted data packets. As mentioned before, both use convolutional codes and block check sequences of varying strengths, so as to produce different coding rates. CS-1 is based on the ½ rate convolutional code, whereas CS-2 is punctured to obtain an approximate rate of 2/3.

In the simulation 9000 packets were generated at 8 kb/s with a 20 ms frame length. The IR, IR-DYN and also reference packets are transmitted using the CS-1 coding scheme, which is the more powerful scheme. The compressed packets are transmitted using the CS-2 coding scheme at the same C/I. The previous section demonstrates that the ARUS technique has a significant impact on the PLR performance and the objective quality of the speech/video quality, especially at high bit error rates. In the next sub section, the effects of the prioritisation are investigated and the comparison results are presented.

Figure 4-34 demonstrates the packet loss rate performance against C/I when prioritisation is used with ARUS. By activating the prioritisation, most of the important packets, which bring the compressor and de-compressor into synchronisation, and/or update the de-compressor context, are given priority.

As can be seen in figure 4-34, giving extra priority to more important packets, increases the packet loss rate performance. The cost of this improvement is throughput. The scheme reduces the actual data throughput due to extra channel coding on specific packets. Table 2-3 lists the actual data throughput with different channel coding schemes. Using the CS-1 coding scheme reduces the user data bit-rate by 4.35 kb/s from 13.4 kb/s to 9.05 k/b/s, compared to the CS-2 coding scheme. The difference is equivalent to 32 %, which may make it impractical for use in all scenarios. At 7 dB C/I, the packet loss improvement with prioritisation is 4.21 % compared to standard ROCCO, and 1.21 % compared to ROCCO with ARUS. For better channel conditions, such as 15 dB C/I, the improvement is not as large. The packet loss rate is reduced by 1.06 % and 0.34 % compared to standard ROCCO and ROCCO with ARUS, respectively, at 15 dB.

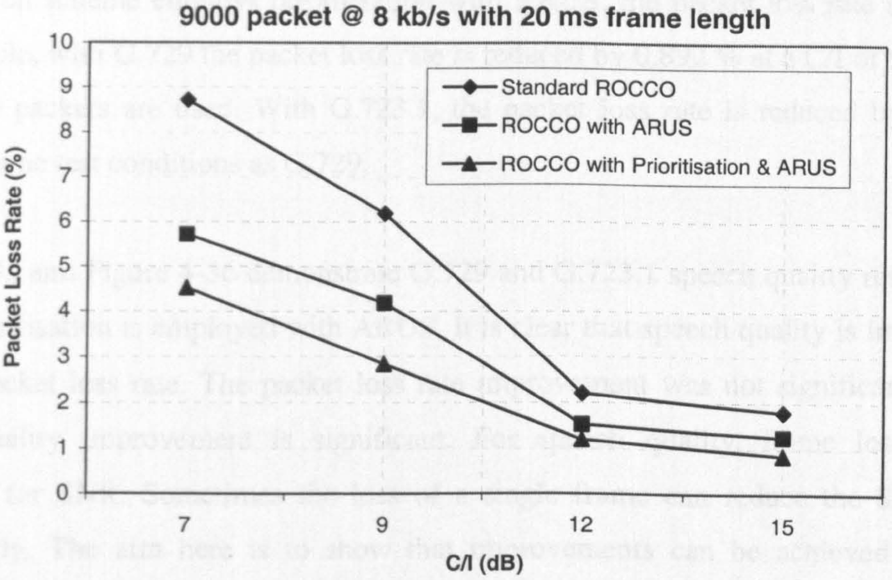


Figure 4-34: Packet Loss Rate performance with Prioritisation + ARUS

4.10.3.1 Speech and Video Codecs Performance

In this section, the packet prioritisation scheme, introduced in section 4.9 is tested with the G.723.1 and G.729 speech codecs and the MPEG-4 video codec. In the speech simulations, 16 seconds of the same speech sample is used with both codecs. Because of the different frame lengths, for G.729 1600 packets are produced, and for G.723.1 codec 533 packets are generated and transmitted.

PACKET LOSS RATE (%)				
C/I (dB)	7 dB	9 dB	12 dB	15 dB
No COMPRESSION	52.183 %	26.306 %	15.860 %	9.175 %
STANDARD ROCCO				
G.729	13.735 %	9.723 %	2.919 %	2.17 %
G.723.1	6.715 %	3.427 %	1.504 %	0.309 %
ROCCO with ARUS				
G.729	8.074 %	5.634 %	2.186 %	1.55 %
G.723.1	4.538 %	2.473 %	1.041 %	0.0802 %
ROCCO with Prioritisation + ARUS				
G.729	6.47 %	4.742 %	1.338 %	0.928 %
G.723.1	4.062 %	2.085 %	0.871 %	0.799 %

Table 4-7: Packet Loss Rate performance with ARUS + Prioritisation

The packet loss rates are listed in Table 4-7 for both applications, and show that when the compression scheme employs prioritisation with ARUS, the packet loss rate is reduced. For example, with G.729 the packet loss rate is reduced by 0.892 % at a C/I of 9 dB when prioritised packets are used. With G.723.1, the packet loss rate is reduced by 0.954 % with the same test conditions as G.729.

Figure 4-35 and Figure 4-36 demonstrate G.729 and G.723.1 speech quality respectively, when prioritisation is employed with ARUS. It is clear that speech quality is improved as well as packet loss rate. The packet loss rate improvement was not significant, but the speech quality improvement is significant. For speech quality, frame loss is very important for SNR. Sometimes the loss of a single frame can reduce the SNR value significantly. The aim here is to show that improvements can be achieved by using different methods of packet transmission. For example, using G.729 at 9 dB C/I, the SNR value improved from 16.78 dB to 29.05. With G.723.1, under the same conditions the SNR improvement is about 10 dB, from 25.12 dB to 14.64 dB.

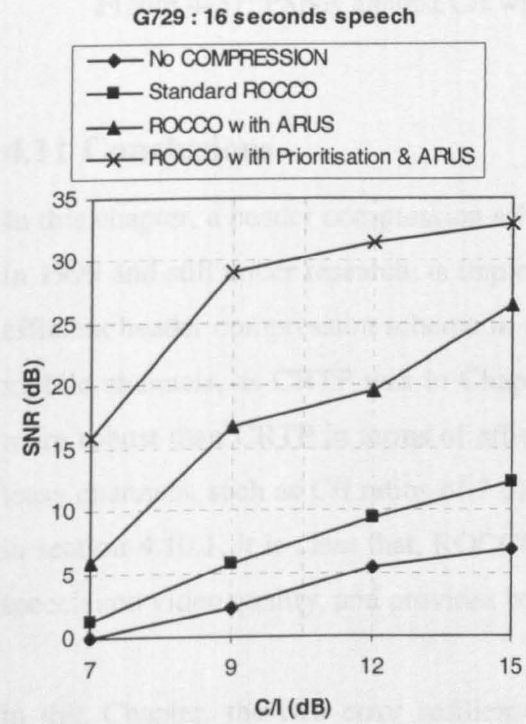


Figure 4-35: G729 SNR with Prio + ARUS

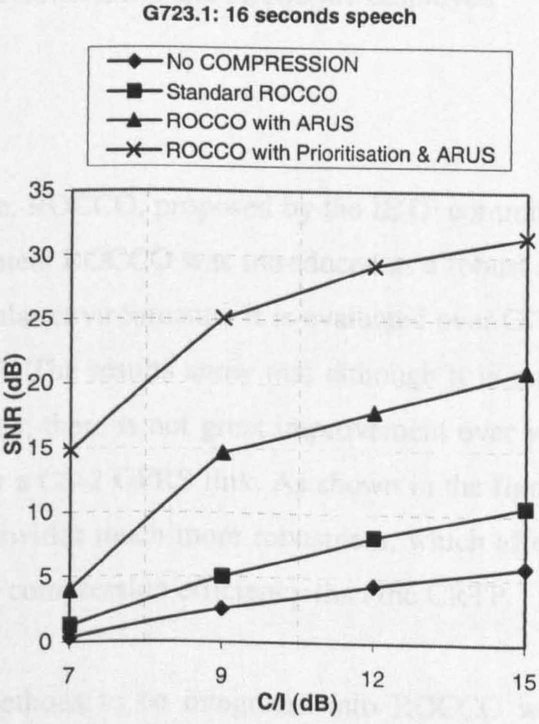


Figure 4-36: G.723.1 SNR with Prio + ARUS

Figure 4-37 shows MPEG-4 video quality when prioritisation is employed with ARUS. It can be seen that, there is a quality improvement even with video applications. At 9 dB C/I, the PSNR improves by 1.47 dB when prioritisation is employed with ARUS.

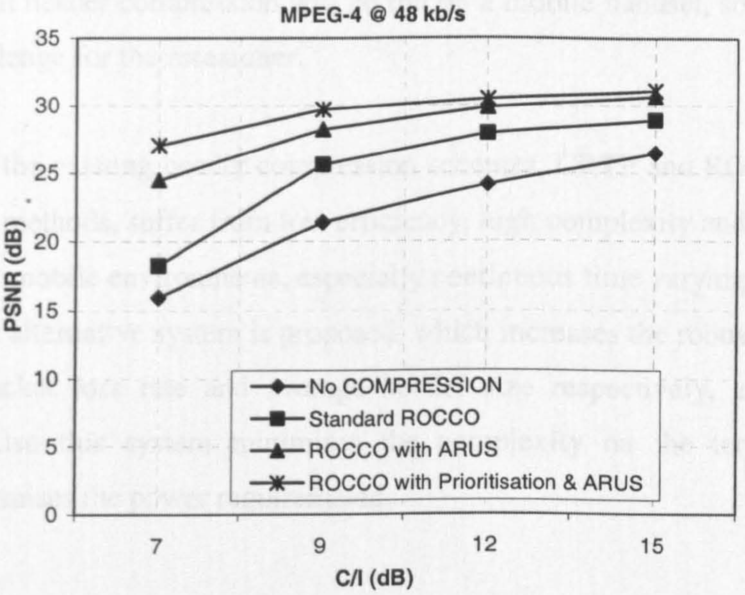


Figure 4-37: PSNR against C/I when Prioritisation and ARUS are employed

4.11 Conclusions

In this chapter, a header compression scheme, ROCCO, proposed by the IETF committee in 1999 and still under research, is implemented. ROCCO was introduced as a robust and efficient header compression scheme in cellular environments. It is evaluated over GPRS mobile channels, as CRTP was in Chapter 3. The results show that although it is much more robust than CRTP in terms of efficiency, there is not great improvement over very lossy channels, such as C/I ratios of 7 dB for a CS-2 GPRS link. As shown in the figures in section 4.10.1, it is clear that, ROCCO provides much more robustness, which affects speech and video quality, and provides better compression efficiency than the CRTP.

In this Chapter, the two error resilience methods to be integrated into ROCCO were introduced by the author. The resulting ROCCO-based schemes were evaluated and compared with standard ROCCO. The results show that these two methods help to improve robustness and reduce the packet loss rate. It is clear that the difference is more significant at lower C/I conditions. However, in terms of efficiency, performance decreases and the average header size increases in every case.

In addition to all these points, ROCCO is significantly more complex than the CRTP. Considering that header compression will be run on a mobile handset, so complexity will be another challenge for the researcher.

To summarise, the existing header compression schemes, CRTP and ROCCO, even with error resilience methods, suffer from low efficiency, high complexity and poor robustness in high bit error mobile environments, especially continuous time varying channels. In the next section, an alternative system is proposed, which increases the robustness, efficiency in terms of packet loss rate and average header size respectively, according to the applications. Also this system minimises the complexity on the terminal and thus effectively minimises the power requirements.

Chapter 5

5 Adaptive-forward Buffering and Header Stripping over Wireless Link

5.1 Concept

The two IETF header compression techniques, CRTP [1] and ROCCO [6], are used to reduce the protocol inefficiency of IP/UDP/RTP. These techniques use lossless compression algorithms, where the result of the de-compression must be bit-by-bit identical with the original compressed header. These schemes, CRTP and ROCCO were implemented and some error resilience methods have been applied on them in Chapters 4 and 5 respectively. However, the results show that for certain real-time applications and radio link characteristics, the presence of even a single octet of header may result in a significant decrease in spectrum efficiency compared to existing circuit-switched technology. ROCCO can compress the protocols down to one octet, in favourable channel conditions. But in cases where corruption occurs, this single compressed header octet could cause the packet to be discarded, which may result in significant degradation in perceptual quality for multimedia applications. Also, one of the main and crucial assumptions for these compression schemes is that the packets are always in their original transmitted (from source) order at the compression point. However, in best-effort packet transmission scenarios, such as that provided by IP networks, data packets are subjected to time-varying delay as a result of differing levels of congestion and varying loads in different parts of the network. Variations in packet delay, also known as jitter, cause VoIP packets to arrive at their destination in uneven patterns. (This can result in packets either arriving out of order, or alternatively with a high degree of variability in packet arrival time. Typically, the solution to jitter problems in mobile access networks is to ensure that mobile terminals are able to buffer the input streams to smooth out variability. This can require significant amounts of memory. In the case of multi-stream transmission, stream synchronisation must also be performed. Even with terminal buffering, there is no guaranteed service, since mobile environments can change very frequently, and cannot be predicted in advance. Also, any unexpected error in the header will cause extra processing time delay at the de-compressor, which can be very serious. In addition, the higher degree of variability introduced by the radio access network reduces the efficiency of ROCCO schemes significantly, even though it provides reasonable robustness.

Therefore, the average required overhead can never be one byte due to the following:

- The time-varying characteristics of mobile links
- The required overheads during the initialisation process
- The Core Network transmission delay or the variability of packet arrival times

In the light of these facts, Chapter 5 presents a novel system, "Adaptive Time-Windowing (ATW) and Adaptive Forward-Buffering (AFB)", which minimises header sizes, conserving bandwidth and radio spectrum. It also minimises the effects of varying transmission delay. Within this scheme, an "Application-Defined Packet (ADP)" and a "Smart Packet (SP)" are proposed, which improve robustness and efficiency.

The proposed scheme makes use of the content-specific information related to a bit-stream that is supported by the standard applications (e.g. types of video codec or speech codec), to overcome the above problems. The main concept behind the reduction of header overheads is based on the provision of smooth play-out of packets over the air interface in the downlink. This minimises the computational complexity and power requirements on the handset by providing 0-byte header compression. In addition, the processing delay that is used to reconstruct the header is omitted. The main advantage of this scheme is that compression of consecutive packet headers is independent of each other over wireless links, unlike when using either the CRTP or ROCCO schemes. Therefore, there is no risk of consecutive packets being discarded due to corrupted header information. One of the essential requirements of this scheme is to place an Edge Gateway (Proxy) at the Edge of the Core Network (EoCN) between end-user terminal and

CN. This proxy has different functions for uplink and downlink transmission, which are explained in detail in the following sub-sections.

This Chapter is organised as follows: The Second section presents an overview of the expected Internet traffic model. The third Section discuses the importance of using the reverse-proxy at the edge of the core network, which is the RNC (Radio Network Controller) in UMTS, and also the proxies functionality for real-time multimedia communications. Also it is focused on adaptive-time windowing on the uplink and adaptive-forward buffering techniques on the downlink, respectively. The proposed mathematical model of the core network, priority queuing and buffer size optimisation schemes are presented. In addition, the ADP and SP functions are explained and their formats are presented. In the fourth Section, the signalling process for call set-up (call initialisation) and call termination is explained. Section six describes the computer simulations and their results. Finally, Section seven draws the concluding remarks.

5.2 Internet Multimedia Traffic Modelling

In packet-switched technology, end systems (mobile terminals in this case) are not directly attached to each other via a single link. Instead, they are indirectly connected to each other through intermediate switching devices known as routers. A router takes information arriving on one of its incoming communication links and then forwards that information on one of its outgoing communication links [139]. The path that transmitted information takes from the sending end system through a series of communications links and routers to the receiving end system, are known as a route or path through the network. A core network consists of a collection of routers connected together by transmission links [137] [140].

Figure 5-1 shows the basic configuration model of a single router within the core network. The configuration shows streams of traffic (packets) emanating from a number of heterogeneous sources being multiplexed by a first in first out (FIFO) or first come first serve (FCFS) buffer with infinite capacity. The buffer is served by a high capacity server such that no packets are lost during transmission

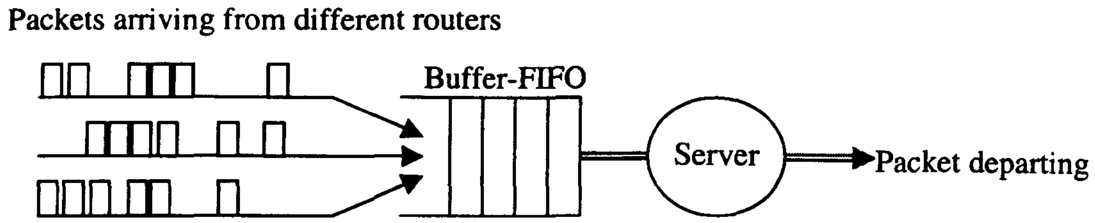


Figure 5-1: Model of single router and queue

The core network traffic is normally generated by a very large pool of users using applications that are uncoordinated in advance. The time that the packet spends from arrival to departure, called waiting time, varies according to the server processing time, packet length, and the number of packets that the buffer contains when a packet arrives. These factors cannot be predicted in advance, and hence the end-to-end delay of each transmitted packet can vary within a wide range. These are very problematic issues for real-time traffic, usually based on UDP as the transport protocol. This type of traffic is incapable of varying its flow rate when faced with changes in delay and throughput across the network. Misordering of packets reduces CRTP and ROCCO header compression schemes' performances, and also the QoS for the applications are severely affected.

5.2.1 Pareto Distribution

In the light of these facts, the traffic model is designed and evaluated based on the previously described characteristics of the core network. The traffic characteristics of interest include packet arrival rate, inter-arrival time, burstiness, and distribution of arrival times between packets. The temporal relation between sources is especially important in multimedia traffic. A good model is one that can accurately predict the statistics of the modelled system. From analytical results [124] it has been found that the Pareto distribution is an appropriate model to represent the core-network Internet model for one-way delay distribution as well as Round Trip Time (RTT) distribution [124]. This often results in packets either arriving out of order, or alternatively with a high degree of variability in packet arrival time.

$$F(x) = 1 - (\beta/x)^{\alpha}\cdot\ x >= k$$ Equation 5.1

The core network was modelled using OPNET with a Pareto distribution model (Equation 5.1). The probability distribution function (pdf) is defined as,

$$p(x) = \frac{\alpha\beta^{\alpha}}{x^{\alpha+1}} \qquad \textit{for}\ \alpha,\ \beta > 0,\ x > \beta \qquad \text{Equation 5.2}$$

The mean and variance of the Pareto distribution is given respectively by

$$\mu = \frac{\alpha\beta}{\alpha - 1} \qquad \text{Equation 5.3}$$

and,

$$\sigma^2 = \frac{\alpha\beta^{2}}{(\alpha-1)^2(\alpha-2)} \qquad \text{Equation 5.4}$$

where, α is the shape parameter and β the location parameter, which sets the minimum core network delay, and x is the random variable, which varies with time. So, $\alpha > 2$ for this distribution to have a finite mean and variance. The Pareto distribution is a distribution with an infinite variance whose distribution has an infinite variance implying that the variable can take on extremely large values with non-negligible probability. It represents the core network delay, which includes packet arrival rate, inter-arrival time, burstiness, and distribution of arrival times between packets.

5.2.2 G/M/m Model

The studied system is G/M/m model, which represents exponential service time, but allow the packet arrival pattern to be quite general. The inter arrival pattern is defined by the distribution of inter arrival times. This model implies that the packets arrive in burst, meaning that for short periods of time packets arrive randomly, and then there is an extended period during which no packets arrive and then another burst occurs. Here, $A(t_n)$ is used for an arbitrary inter-arrival time distribution $A(t_n)$, represented by G and $S(t_n)$ is used for an exponentially distributed service time, which is equal to the application frame rate, represented by M and "m" represent the number of servers that are used, which for the single server system, m = "1".

5.3 Reverse-Proxy

In UMTS [131] [135], every Packet Service (PS) domain RAB (Radio Access Bearer) is associated with one RB (Radio Bearer), which in turn is associated with one PDCP entity. The PDCP entities are allocated in the PDCP sub-layer. Several PDCP entities may be defined for a mobile terminal (called User Equipment in third generation network) with each using the same or different protocol PDCP sub-layers [15]. If the application does not tolerate any information loss, the PDCP maintains sequence numbers for the Service Data Units (SDUs). SDU is the payload data unit within 3G, and is transported by the Radio Access Bearer Service. Thus, different radio bearer channels could be used for different applications. This one-to-one mapping between service flow and channel maximises the spectrum efficiency and also minimises the Packet Loss Rate due to the corrupted compressed/uncompressed header information. By taking full advantage of advanced multiplexing techniques, these channels may be shared by different users within the same cell. The delay between the UE and the core network (CN) is significantly low and mis-ordering of packets is not expected. So, by taking full advantage of the provided information related to a bitstream, and the synchronous nature of packets sent over the air interface, the packets can be transmitted over the air interface on an allocated radio access bearer (traffic channels) without any IP/UDP/RTP headers.

To do this, a new edge gateway (Reverse Proxy) is proposed in this thesis. These reverse proxies are placed at the Edge of the Core Network (EoCN) which is the Radio Network Controller (RNC) point. These edge proxies break the mobile-to-mobile communication path into three parts; transmitting-link (uplink), core network and receiving end-link (downlink) (see Figure 5-10). In the overall application system, these links do not depend on each other, and an error in one of the links does not affect the performance of the other transmission links. Unlike the usual use of web proxies in IP networks, where a network node close to the terminal duplicates the information located in some geographically-remote server, these nodes act as proxies for terminals (or clients). They therefore act in the reverse-direction to traditional proxy servers.

The Edge Proxy has two main responsibilities. The first is to generate all required protocol header data, such as destination/source addresses, destination/source port numbers, payload size, sequence number, timestamps, and data type for all media packets on the uplink. The second responsibility is the removal of all header information for out going packets on the downlink. Thus, to ensure compatibility with radio links and core network transport (IP) protocols, an edge-gateway (reverse proxy), provides Adaptive time-windowing on the uplink and adaptive forward buffering on the downlink.

During the call set-up procedure, required information, such as IP address, and port numbers (source and destination) are transferred to the Edge Proxy by the link layer. This can be achieved transparently over the link layer. At the same time, the two end-to-end edge-proxies synchronise and transfer the service and user information to each other, then establish the call. Once the call is established, only the payloads are transmitted over the radio links, since the edge-proxies have all the required information about the service and the users. It is the edge-proxies responsibility to maintain synchronisation with each other, and also between the mobile terminals that they are connected to. They are responsible for generating an adaptive table to store the transferred information once the call is established.

On the uplink, the full headers (IP/UDP/RTP) are encapsulated at the Reverse-Proxy as if the packets are transmitted from the Edge Proxy. So, packets will have full header information within the core network until they reach other end of the core network. On the downlink, the overhead protocols are fully removed at the edge of core network, and only the payload is transmitted over the wireless link to the end user. To achieve, this two techniques are proposed for use within the Edge Proxy: adaptive-time windowing for the uplink, and adaptive-forward buffering for the downlink, which are explained in detail in sub-sections 5.3.1 and 5.3.2 respectively. In addition, two different packet formats are proposed for use within the Edge Proxy on the downlink, which are called smart packet and application-defined packet. The smart packet format is explained in section 5.3.3 and in section 5.3.4, the application-defined packet format is described.

5.3.1 Adaptive-time Windowing Technique

The Adaptive-time windowing technique is one of the Edge Proxy's built-in functions and is also the only function that is used on the mobile terminal for the proposed scheme. It is used to detect the packet loss rate over radio channels. The Edge Proxy implementation is used for incoming packets on the uplink, whiles the mobile terminal version, is used on the downlink for packets transmitted from proxy to end-user. Although, mis-ordered packet arrivals are not expected over radio links, however packet loss may occur.

The detection of packet losses over the wireless channels is very important for ensuring that the correct protocol information is attached to the payload, especially timestamps and sequence numbers. Any incorrect protocol information attached to the payload and transferred through the core network will propagate through consecutive packets. For instance, on the uplink the Edge Proxy takes all incoming packets, and generates the protocol headers, including the timestamp and the sequence number on them assuming no packet loss. In case of packet loss, the errors will propagate and will cause quality degradation for the end user. The same problem may occur on the downlink, and hence the synchronisation between the applications will be lost. Therefore, to overcome this problem, a time-windowing technique is used to detect packet loss on the uplink and downlink to secure correct protocol encapsulation and synchronisation based on the information it has.

The only problem is to identify the beginning of a talk spurt for speech and the start of a video frame. To overcome this problem, a smart-packet is proposed, which is used to inform the receiving end when a talk-spurt for speech begins and when a new frame starts for video. The Smart Packet and its format are described in the following sub-section 5.6. This sub section continues with a description of the advanced windowing technique

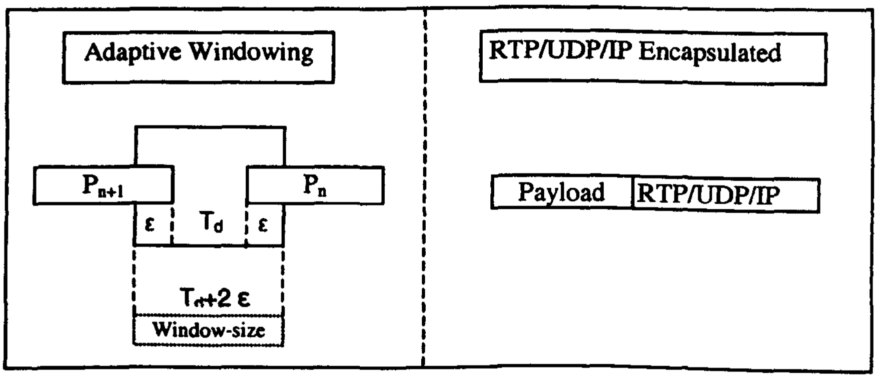


Figure 5-2: Adaptive-Time Windowing Technique

T_f = Application Frame-rate

$\varepsilon = aT_{f,}$ where $0 < a < 1$

$P_n = n^{th}$ packet's arrival time

$P_{n+1} = (n+1)^{th}$ packet's arrival time

$T_d = P_n - P_{n-1}$ = consecutive packet time difference

$T = T_d + 2\varepsilon$

$$(T_d - \varepsilon) < T_d < (T_d + \varepsilon) \qquad \text{Equation 5.5}$$

Referring to Figure 5-2, the time "T_d" between P_n, and P_{n+1} is measured. If it is less than a predetermined window size T, a header is added. Generally T_d is equal to the service frame rate "T_f". If the inter packet time "T_d" is greater that "2T– ε", a packet is regarded as missing. The length of the period, T, will be the expected inter packet time "T_d" plus a margin "ε", the length of which can be varied in an adaptive manner depending on the application and the air-link delay. Once the lost packet is detected, the uplink Edge Proxy either:

- Continue to forward only incoming media packets, but with corrected header information to inform the downlink Edge Proxy that a packet was indeed lost or it can insert an ADP as a substitute for the lost packet.

The ADP can be used for a variety of purposes depending upon where it is used. This procedure may be implemented in either the up-link Edge Proxy or the downlink-receiving terminal.

In either case the negative effect of the packet loss on the quality may be reduced without losing synchronisation between two or more end users. This scheme does not require the use of a feedback channel nor retransmission for real-time services.

5.3.1.1 Service dependent

The window size is purely application dependent. The window size is set according to the application characteristics, during the call set-up negotiation process. Generally, window-size is approximately the same as the packet transmission rate. The window-size changes adaptively according to the service codec that is used, as long as it gets acknowledgement from the sender.

5.3.2 Adaptive-forward Buffering Technique

As described in earlier sections, as a result of the best-effort nature of IP-packet communications, packets can have significant variations in arrival time that will either cause packets to arrive out of order, or alternatively with a high degree of arrival-time jitter. This will degrade the output quality of applications, and in particular, the intelligibility of speech services will be significantly affected. To provide a high quality real time communication, the packet loss rate should be kept small. However, the characteristics of the CN will always have a significant negative impact upon the quality and spectrum efficiency performance, particularly when header compression is used over radio link, such as CRTP and ROCCO. The reason for this is the time-varying delay, which means that some packets arrive to the end user after play-out time, which must be discarded. This will have degrade the decoded media quality. Moreover this variation will degrade the efficiency of CRTP and ROCCO compression algorithms.

To overcome these problems, the Adaptive-Forward Buffering technique is proposed in this section, which is located at the Edge of the Core Network (EoCN). Primarily, it plays a significant role in ensuring that radio spectrum is used with maximum efficiency on the downlink. It receives the packets from the core network which have experienced varying transmit delay and may well be mis-ordered. Its main function is to provide 0-byte header transmission over the radio link. In order to eliminate the jitter effect, it buffers a number of packets before it starts to transmit them to the end user. It re-organises the received packets into their original transmission order and smooth out the variation of the arrival time. This is achieved by transmitting packets at regular intervals according to the source application frame rate. All headers are stripped, without any headers as if the packets are generated at the Edge Proxy. It always provides a priority packet service, and transmits the packets in their original transmission order, regardless of their arrival time.

The Forward Buffer size has to be adaptive based-on the application as well as the delay statistics of the packet arrival time. The initial buffer size is important, particularly for real-time applications, which is explained in the subsection 5.3.2.2. Too long a buffer size will result in an excessively large delay, whereas too short a buffer will result in too many discarded packets. The initial buffer size is set by considering the maximum affordable delay according to the application. Effectively, it minimises the computational complexity, power and memory requirement on the handset.

In order to overcome buffering problems, the downlink Edge Proxy includes an adaptive forward buffer as shown in Figure 5.7. The main functions of the adaptive forward buffer are to achieve 0-byte header transmission over the radio link, and minimise the complexity and memory requirements on the handset. It holds a number, *N*, of the received packets P_1 to P_n and re-organises them into their original transmission order and transmits the packet traffic smoothly based on the frame rate of application. The forward buffer has to be adaptive so that the number of packets it can hold can be varied depending on the application as well as delay statistics of the packet arrival time. The arrival times must therefore be continuously monitored and the delay adjused accordingly. This is because the maximum delay limit used will be different for different services.

Adaptive buffering allows smoothing out of the variability of packets, allowing as many packets as possible to reach the end user decoder in playout-time, while keeping the buffering delay as low as possible. The size of the buffer and play-out time are highly dependent on the delay distribution of the packets. Even if the forward buffer is adaptive, there can still be packets that do not arrive at the Edge Proxy (buffer) on time. In this scenario, "Application Defined Packets" are inserted of those packets so as to manage the adaptive forward buffer more efficiently and smooth out the variability.

5.3.2.1 Adaptive Priority Queuing for G/M/1 Model

Normally the order in which packets are served in a router is on a first-come-first-served basis everywhere in the core network. In many cases, this is the actual order in which packets are originally served. This places responsibility for reordering packets upon the terminal. This also reduces header compression scheme's protocol efficiency and thus increases the packet loss rate, as well as complexity and power requirements on the handset.

Therefore, proposed schemes involving other disciplines, such as the random selection of packets for service in the waiting line, or the probability that an arriving customer will have to wait for service, are the same for a random queue discipline as they are for a FIFS discipline. It is only when priorities are assigned to the arriving customers on a basis such as their expected service time and/or cost of waiting in line, that the system parameters of interest are affected. This is potentially one of the most important contributions that queuing theory can make

5.3.2.2 Buffer Design and Size Optimisation Algorithm

The initial buffer[††] size setting is very crucial for the proposed system. This is set according to the arrival time variation, buffer service rate and affordable delay (max_delay) based on the applications.

d_T = Transmission delay

P_n = Initial number of packets in the buffer

λ_d = packet delivery rate

$$\text{max\_delay} = d_T + (P_n * \lambda_d) \qquad \text{Equation 5.6}$$

Optimised initial packet numbers in the buffer:

$$P_n = (\text{max\_delay} - d_T)/\lambda_d \qquad \text{Equation 5.7}$$

5.3.2.3 Waiting Time distribution

The Waiting time is defined as the total time that packets spend in the buffer. When a priority service that is carrying out re-ordering is used, the waiting-time between packets can be significantly different. Because the considered case here is not of the first-in-first-out discipline, the arrivals sent earlier by the sender will have some priority over others. The mean waiting time and average number of packets in the buffer will remain unchanged. However, as shown in Figure 5-3, it is obvious that some arrivals are served sooner than they would have been for the first-in-first-served discipline, and likewise some arrivals wait longer. Figure 5-3 shows the waiting-time of some packets in the buffer obtained from simulation.

[††] Initial buffer size represents number of packets in the buffer, just before the communication start.

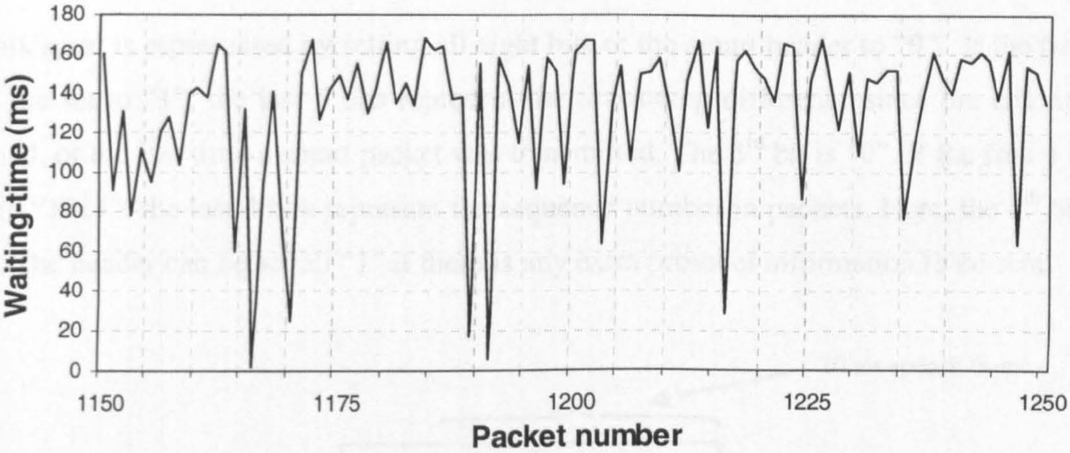


Figure 5-3: Waiting-time of packets between 1150th -1250th in the buffer

5.3.3 Smart Header

The Smart Header is a new packet type, proposed in this thesis. It is used to support packet transmission over wireless links with 0-byte headers. The concept of the smart header and its format is explained in the following sub section. It is used across the mobile link and introduces one extra byte on top of the payload, whenever it is used.

5.3.3.1 Smart Header Concept

By using the proposed scheme, packets are transmitted with 0-byte headers across the mobile link, which minimises the bandwidth requirement. Although the packet transmission delay over radio link is negligible, but still the packet loss cannot be prevented. Because of this the receiver end on the uplink or downlink needs to identify when a talk spurt starts for speech applications and when a new frame starts for video applications, otherwise the two ends can lose synchronisation. To prevent this, Smart Packets are used to notify the receiving ends when a silence period is finished for speech and when a new frame starts for video. This retains synchronisation between mobile terminal and ULeP as well as the downlink Edge Proxy and the end user. The smart header is also used for timestamp and sequence number checks at the ULeP when the headers are generated. It is introduced as an extra one byte overhead on top of the payload. It is used when a talk spurt starts and every "T" ms after a talk spurt, which can vary between 200-400 ms (for the applications with 20 ms frame rate, T = 300 ms which is every n*15th frame after talk spurt).

5.3.3.2 Smart Header Format

A talk spurt is represented by setting all eight bits of the smart header to "0". If the first 2 bits are set to "1", the last 5 bits represent the timestamp difference since the talk spurt started, or the last time a smart packet was transmitted. The 3rd bit is "0". If the first 4 bits set to "1011", the last 4 bits represent the sequence number in packets. Here, the 4th bit is "0". The header can be set all "1" if there is any extra protocol information to be sent.

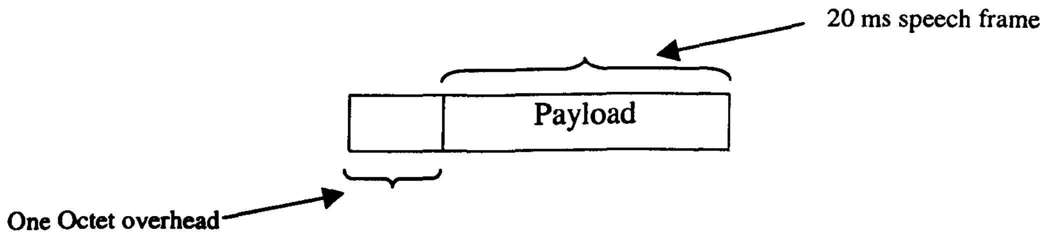


Figure 5-4: Smart Header

00000000	: smart header
110	: smart header with timestamp difference
1011	: smart header with sequence number

0	1	2	3	4	5	6	7
8-bit synchronisation word							
Speech/video payload							

8-bit synchronisation word format are shown blow;

When talk spurt or new frame start;

0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0

smart header with timestamp difference;

0	1	2	3	4	5	6	7
1	1	0	Timestamp Dif.				

smart header with sequence number;

0	1	2	3	4	5	6	7
1	0	1	1	Seq. Number dif.			

5.3.4 Application-defined Packet

The *Application-Defined Packet* is a new packet type, which is included in the proposed system in this thesis. It is used to support service quality by providing extra information for the decoder application, and is employed across the mobile link. The concept of the application-defined packet and its format are explained in the following sub section.

5.3.4.1 ADP Concept

Adaptive-forward buffering allows smoothing out of variability in packet ordering to allow as many packets as possible to reach the end user decoder within playtime, whilst keeping the buffering delay as low as possible. The size of the buffer and the play-out time are highly dependent on the delay distribution of the packets. Even if the forward buffer is adaptive, there may still be packets that do not arrive at the Edge Proxy (buffer) on time and are therefore discarded. In this case, ***"application-defined packets"*** are inserted instead. The ***"application-defined packets"*** are used to manage the adaptive forward buffer more efficiently and smooth out the variability. Basically they act as a resynchronisation marker.

Alternatively, the application-defined packet can contain the reconstructed or estimated payload inserted instead of the missing packets. This provides as much information as possible to enhance the subjective quality of the application as well as perceptual quality for the end user. The negative effect of packet loss to the quality is minimised without losing synchronisation between two or more end users. This scheme does not a require feedback channel and avoids re-transmission for real-time services.

Thus the application-defined packet can be used for a variety of purposes depending upon the location where it is used. It can be used on the downlink Edge Proxy. The ADP packets provide enough information to inform the end user that there is a missing packet.

5.3.4.2 ADP format

There are two different formats for *Application-Defined Packets*. The first one is used to send an acknowledgement to the decoder that there is a missing frame, and that the current packet only includes the missing frame sequence number (SN). The second format carries a reconstructed information dataset for the missing frame as well as its sequence number. In both formats, the application-defined packet is represented by a 16-bit resynchronisation marker, which is always defined during the call set-up. It is the first two bytes of the packet.

The total length of the first packet type is three bytes, and it is indicated by setting the first 4 bits to '0' after the resynchronisation marker. The last four header bits represent the four least significant bits of the sequence number of the missing frame. The format is shown in Figure 5-5.

0	1	2	3	4	5	6	7
16-bit resynchronisation codeword							
4-bit flag							

Figure 5-5: The first option of the ADP format

0	1	2	3	4	5	6	7
16-bit resynchronisation codeword							
1	1	1	1	Data length			
Reconstructed speech/video data							
0	0	0	0	4-bit LSB SN			

Figure 5-6: The second option of the ADP format

Figure 5-7 shows the proposed edge proxy with all functions used on the downlink. The second format for ADP includes a resynchronisation code, plus the length of the dataset, which represents the missing frame. It also includes one extra byte, which is placed just before the dataset. The dataset represents the missing frame, and is extracted at the reverse-proxy from the previous frame, which has been sent, and the next frame, which is in the buffer. The reconstructed dataset is placed between the codeword and the last byte.

The extra byte, which is placed in front of the extracted frame, is split into two parts with 4 bits in each part. The first 4-bits are set to "1", to indicate that the ADP includes the extracted dataset instead of a missing frame. The following 4 bits confirm the length of the reconstructed frame. The last byte is as-is in the first option. The format is shown in Figure 5-6.

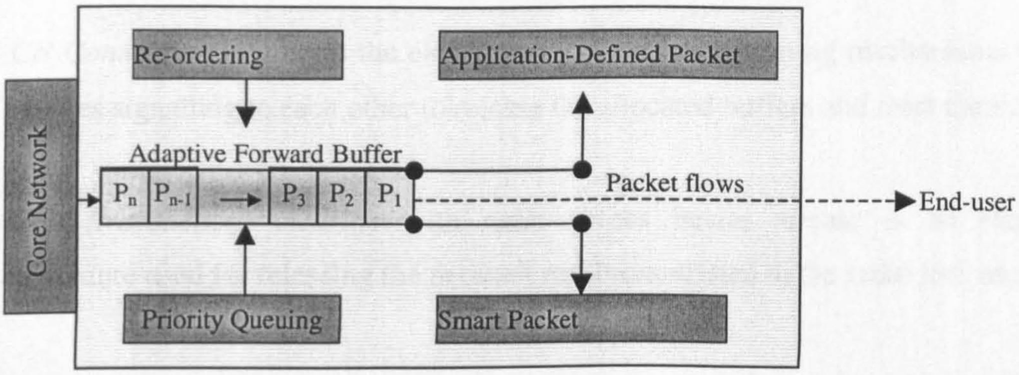


Figure 5-7: Edge proxy for downlink

5.4 Signalling Process

In this section, the model for overall mobile-to-mobile system signalling is proposed and presented by modelling each system procedure as a kind of communication-intensive transaction. The concept of transaction is used in order to underline the fact that these system-wide procedures are precisely defined and that subsequent procedures are fairly independent communication type.

The mobile-to-mobile transaction can be divided into 6 main steps as presented in Figure 5-10. For each part an elementary procedure can be described as follows;

- *Radio Link connection set-up* is the elementary procedure containing activities and message flows to establish a radio control connection between the terminal and the edge-proxies.

- *CN connection set-up* is the elementary procedure where the edge proxies indicate to each other what type of transaction the terminal is requesting. Based on the reasoning information, such as core network condition and the kind of transaction request, the edge-proxies may decide how to proceed with the transaction and initialise the windowing size, as well as the forward buffer with departure rate, or they may decide to terminate the execution of the transaction.

- *Call (Transaction) set-up* with radio access bearer allocation is an elementary procedure, which allocates the actual communication resources for the transaction.

- *CN Connection release* is the elementary procedure containing mechanisms with the proxies signalling to each other to release the allocated buffers and reset them.

- *Call (transaction) clearing* with radio access bearer release is an elementary procedure used for releasing the network resources related to the radio link transaction

- *Radio Link connection release* is an elementary procedure containing mechanisms that operate when the radio control connection between the UE and the access network is released.

Figure 5.10, demonstrates the end-to-end, mobile-to-mobile communication scenario and Figure 5.8 shows the process for signalling and transmitting data over the system. A wireless access communications network comprises a mobile handset "A", and a second mobile handset "B", a fixed core network (CN), a first edge-of-core gateway in the form of an up-link edge proxy (ULeP) and a second edge-of-core gateway in the form of a downlink edge-proxy (DLeP). The description of the system will refer to a one-way transmission with one up-link proxy and one downlink proxy. However it will be appreciated that, for two-way communication, each of the edge-of-core network gateways will have the capability to up-load and download packets, and will therefore each have the properties of both the up-link Edge Proxy (ULeP) and the downlink Edge Proxy (DLeP).

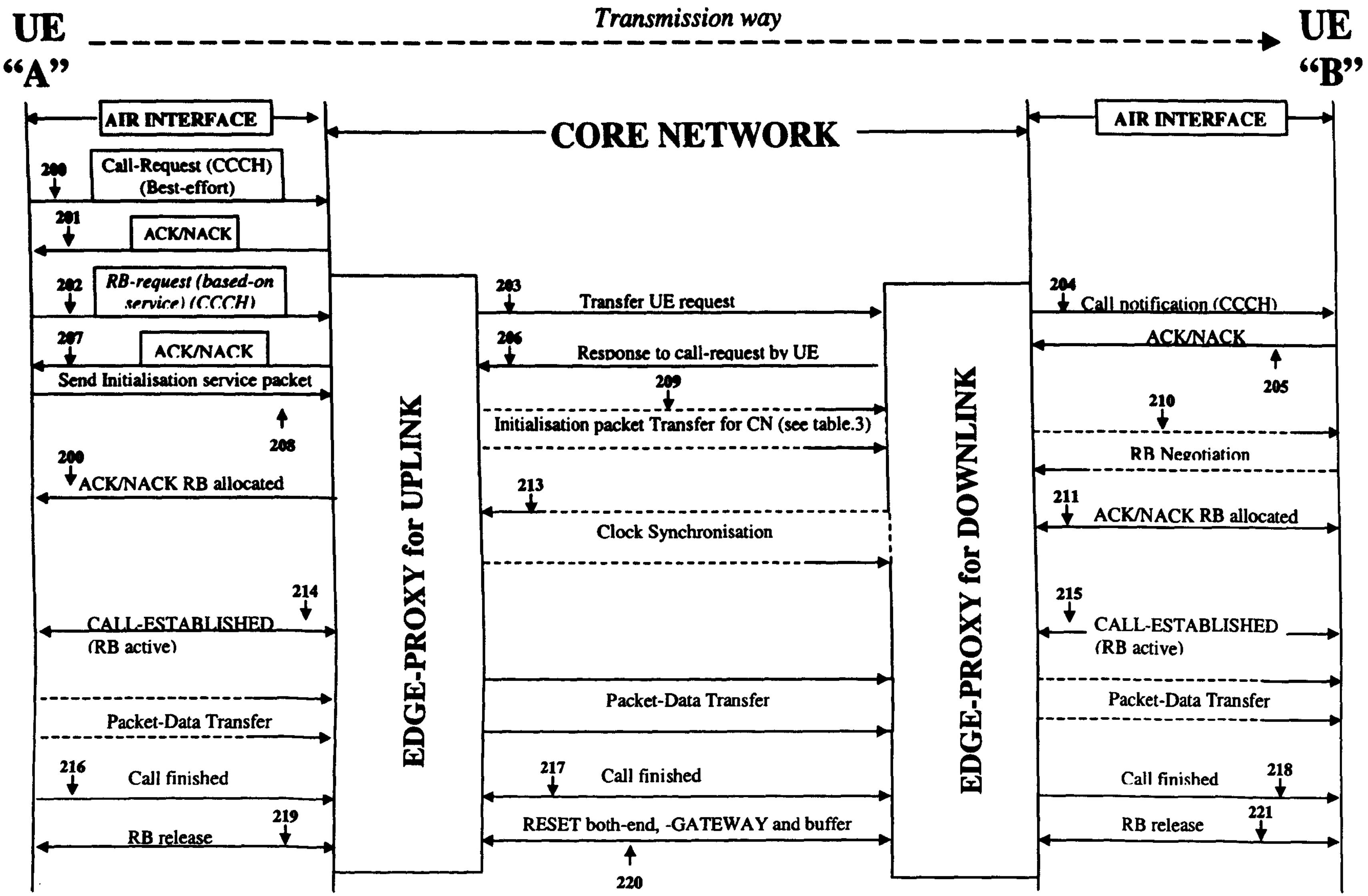


Figure 5-8: Process of Proposed Scheme for Transferring Packet over the System

The system operates as follows. To start communication between the handsets, terminal 'A' and terminal 'B', the first handset sends out a request for a radio link to the ULeP, step 200. This request is acknowledged by the ULeP, step 201. The radio uplink (R-UL), between terminals 'A' to the ULeP is established after the handset 'A' requests a RB, step 202. In this request, terminal 'A' transmits to the ULeP the address of the intended recipient that it wants to set-up a communication links with, in this case the second handset 'B'. This request is then transferred across the CN to the downlink Edge Proxy (DLeP), step 203, which sends a call notification signal, step 204, to the second handset 'B'. The second handset 'B' sends an acknowledgment signal in step 205. When the radio downlink (R-DL) is established, and the DLeP acknowledges the ULeP, then confirmation is sent back to the first handset 'A' by the ULeP, steps 206 and 207 respectively. Then terminal 'A' sends all required protocol information to the ULeP with an "initialisation service packet", at step 208, and the ULeP transfers this packet to the DLeP. The DLeP then communicates via RB negotiation signals with the handset 'B', step 209, which results in a radio bearer (RB) between DLeP and terminal 'B' being allocated in step 210 and an ACK/NACK at step 211. The necessary communication protocols are fully established using this method. For example, all the required information for the applications, such as IP address, port numbers, and also timestamp changes will be transferred to ULeP, DLeP by the link layers.

Once both edge proxies ULeP and DLeP have all the required information at both ends the radio links are established, they synchronise clocks and transfer service and user information. Then both edge proxies establish the call between each link. Step 214 establishes a call between terminal 'A' and ULeP, and step 215 between the terminal 'B' and the DLeP.

After the call has been established, the first handset 'A' transmits over the R-UL a series of packets of data with no headers attached. When the ULeP receives the packets, it generates and attaches headers to each of them. It is assumed that there is very little variability in the delay of a radio link, the order in which the packets are received will be the same as the order in which they were transmitted by terminal 'A'. Packet loss is detected using adaptive-time windowing, whose window size is initialised when the "initialisation service packet" is received by the ULeP.

The ULeP then relays the data, with the headers attached, to the DLeP over the CN using the appropriate network protocols. During transmission over the CN, the packets are subjected to differing delays and therefore the order in which they arrive at the DLeP is not always the same as when they are sent by the ULeP. The DLeP therefore includes an adaptive forward buffer, which puts the packets in a prioritised queue before transmitting them over the R-DL. It enables the packets to be re-ordered on the basis of the RTP sequence number and also the timestamp, so that they are back in the original order. Once the transmission is started, each packet arrives from the CN at the buffer at the DLeP, and is checked with the other packets, which are in the queue. If it was transmitted before any of the other buffer packets by the ULeP, it is given priority and placed just in front of that packet. The headers are then removed from the packets, and the packets are transmitted over the R-DL to the second handset 'B'. Because the second handset 'B' has all the protocol information, and the packets arrive in the correct order over the R-DL, handset 'B' can use data in the intended application without the explicit need for headers from the received data. By using adaptive-time windowing at the link layer, packets are passed to the application layer smoothly. The network layer may extract all of the required information (such as timestamp and sequence number) from the timing of the received data. Alternatively, the original headers may be reconstructed. The only loss here is that the silence duration can be between 50-300 ms longer or shorter. But this is not noticeable for many people.

At this point the call set-up procedure is complete, and call established signals are exchanged between terminal 'A', ULeP, DLeP, and terminal 'B' (step 214 and step 215). Also, the edge-proxies ULeP and DLeP start to transfer the data.

When the call is finished, a call-finished signal, step 216, is transmitted from terminal 'A' to ULeP, and this is transferred to the DLeP from the ULeP, step 217. Once the DLeP has received the call-finished signal, it sends it to terminal 'B', step 218, and at the same time the DLeP sends a call-finished acknowledgment to the ULeP. Then at step 219, the RB between terminal 'A' and ULeP is released at both ends, step 220. At the same time, the RB between the DLeP and terminal 'B' are released, step 221, and a reset signal is also sent between the ULeP and DLeP to reset the gateway and buffer

5.5 Handover Process

There are a lot of questions about handover and its effect on the performance of applications, even in circuit switched networks. In wireless networks handover can occur very often. Handover has to be "seamless" to avoid degradation in quality and to also avoid dropping the calls. With this proposed scheme, there is no need for additional care to be taken; it is more robust because the handover takes place at the Edge Proxy rather than at the RNC.

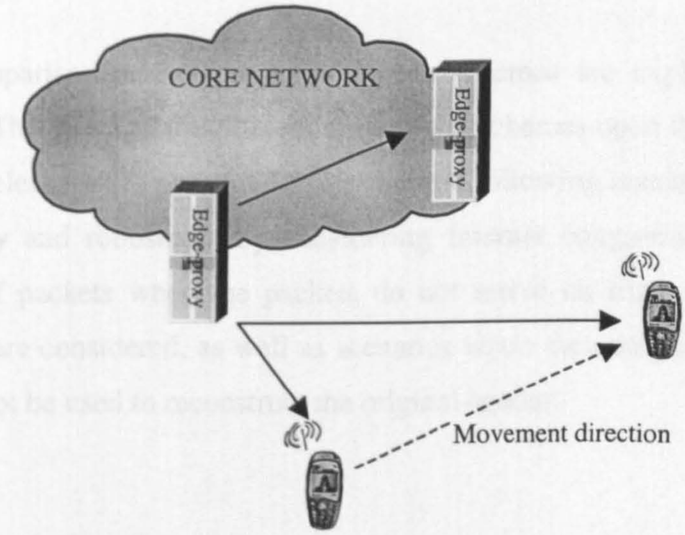


Figure 5-9: Bi-casting

The decoders can afford to lose some data without significantly losing quality. The only information that the application layer needs is how many packets are lost. To further minimise the degradation of voice/video quality, and to support applications that do not tolerate information loss during handover, the bi-casting scheme can be used. Bi-casting involves sending packets to both source and target Edge Proxy during handover preparation. During this time, smart packets are used to transmit sequence information to both source and target neighbouring Edge Proxy. The PDCP also provides a sequence number for the packets, which could be used. Transferring the sequence number can mitigate any discontinuity in the uplink and downlink direction to the target Edge Proxy during handover signalling.

5.6 Simulations and Analysis of the Results

In this section, the simulation scenario is presented. Figure 5-10 is a diagram of the simulated wireless access network, and Figure 5-8 shows the overall signalling required establishing and releasing the call as well as the flow of data in the system of Figure 5-10. The simulation conditions are further detailed within this section. Following that, the results are presented in a comparative way, which clearly identifies the overall performance improvement in terms of the packet loss rate and protocol efficiency with the proposed scheme compared to the IETF standard CRTP and ROCCO when transmitting speech and video.

The comparison and discussion of these schemes are explained in detail within the section. The effect of the header compression schemes upon the speech and video quality over wireless link is presented. Note that the following results focus on the compression efficiency and robustness by considering Internet congestion and the consequent late arrival of packets when the packets do not arrive on time. Packets with errors in the payload are considered, as well as scenarios when the compressed header gets corrupted and cannot be used to reconstruct the original header.

5.6.1 Simulation Scenario

The simulated scenario is shown in Figure 5-10.Speech or video data is produced from the encoder, Mobile Terminal (MT) "A", and is transmitted to the Edge Proxy without any header. At the Edge Proxy for uplink, the RTP/UDP/IP headers are added, and the packets are forwarded for transit through the core network. The headers are removed by the Edge Proxy on the downlink after priority re-ordering and sent to the mobile terminal B over a cellular radio link. At the mobile terminal "B", the headers are reconstructed and the data is passed to the decoder.

The G.729b speech codec was used to compress speech at an output rate of 8kbit/s. The frame rate is 10 ms, resulting in 80-bits/frame. The frames are sent with two-frame-to-one transmission packet mapping, allowing for 160 bits/packet transmissions. During the simulation, 24 seconds of speech (2400 speech frames and 1200 transmission packets), for the video transmission experiments, the MPEG-4 video codec is used at 48kbits/s. The video frames are sent with one-frame-to-one transmission packet mapping.

The core network is modelled with OPNET using a Pareto distribution (Equation 5.2). This is the most appropriate model for one-way distribution. The study uses the G/M/1 model, which is explained in section 5.6.2.

For the time-windowing, T is set to 20 ms and a = 0.1, and as a result ε = 2ms. The packet rate is set to 20 ms, where two-frames are mapped to one packet. The initial buffer size is varied, and the effect is presented in section 5. The wireless links part I and part III, are evaluated using the GPRS access network. The CS-2 coding scheme is employed at a carrier frequency of 1800 MHz, using the TU50 (typical Urban Scenario, mobile terminal velocity of 50kph) Multipath model. Table 4-1 shows the average BERs of the CS-2 code scheme with different Carrier-to-Interference Ratios (dB).

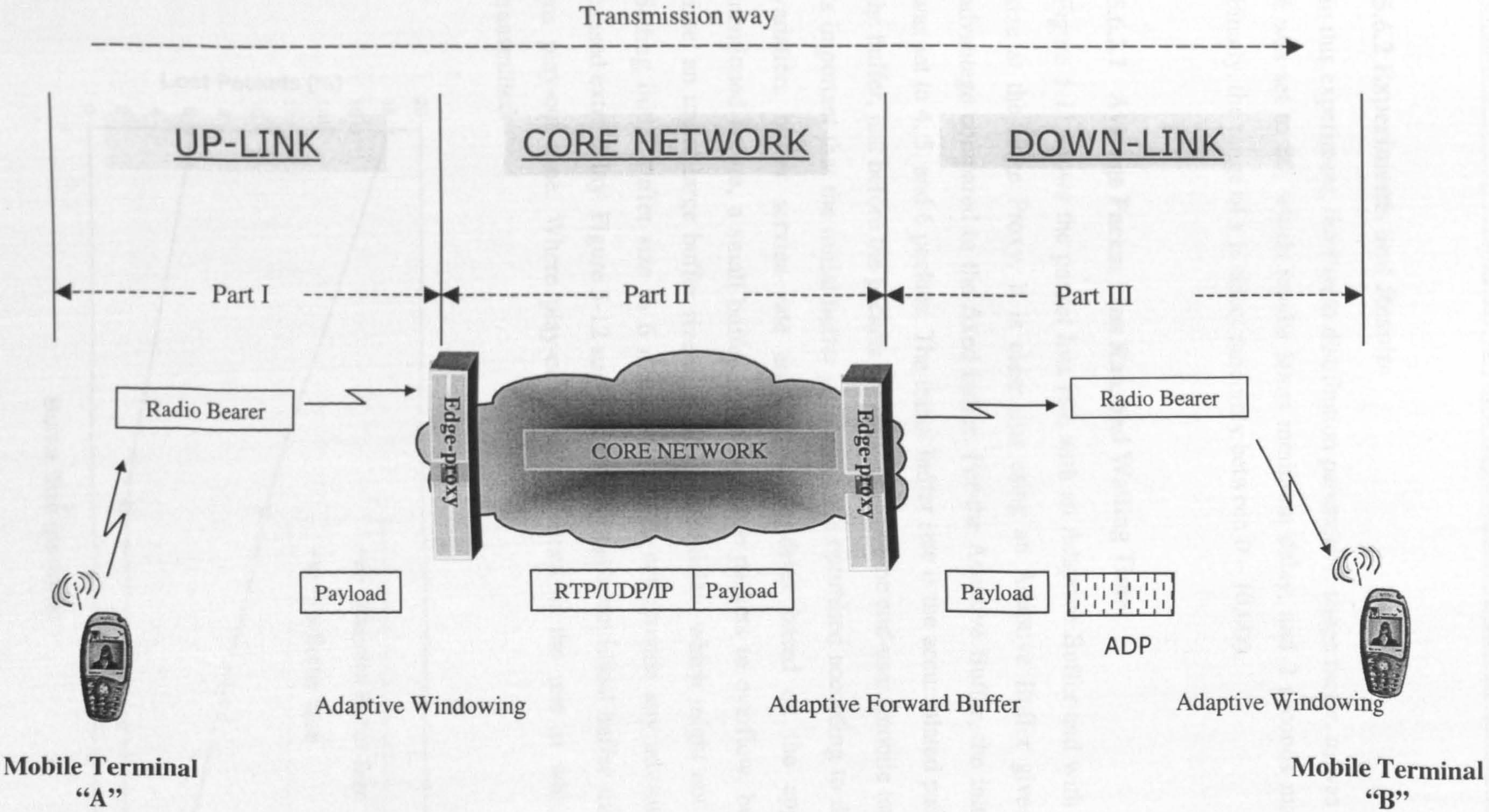


Figure 5-10: Simulation Wireless access Network Scenario

5.6.2 Experiments and Results

In this experiment, the Pareto distribution parameters, shape factor, α, was set to 3.5, and *k* was set to 80, which results 80ms minimum delay, and 2 seconds maximum delay. Finally, the range of *x* is select randomly between 0 – 10,000.

5.6.2.1 Average Packet Loss Rate and Waiting Time

Figure 5-11 shows the packet loss rate with an Adaptive Buffer and with a fixed buffer size at the Edge Proxy. It is clear that using an Adaptive Buffer gives a significant advantage compared to the fixed buffer. For the Adaptive Buffer, the initial buffer size was set to 4, 5, and 6 packets. The initial buffer size is the accumulated packet number in the buffer, just before the packets are transmitted to the end-user (mobile terminal "B"). It is important that the initial buffer size should be optimised according to the arrival time variation, buffer service rate and affordable delay based on the applications. As mentioned before, a small buffer size may cause packets to overflow, but at the same time, an initial large buffer size introduces extra delay, which might not be affordable. Setting initial buffer size to 6 rather than 5 did not provide any advantage, and only caused extra delay. Figure 5-12 suggests that the optimum initial buffer size is 5, with 20 ms play-out time. Where play-out time is defined as the rate at which packets are transmitted.

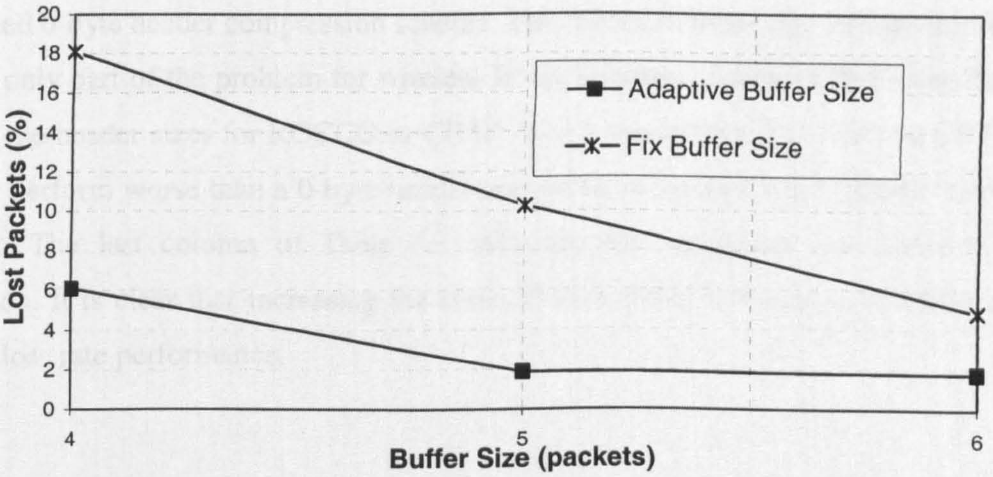


Figure 5-11: Packet Loss Rate within Core network due to buffer overflow and late arrival

Figure 5-12 compares the average waiting time (the time a packet spends in the buffer) for fixed buffer and Adaptive Buffer systems. The results show a clear advantage in using the adaptive scheme. For the optimum initial buffer size, the average waiting time in the Adaptive Buffer is approximately 40ms less than with the fixed buffer scheme.

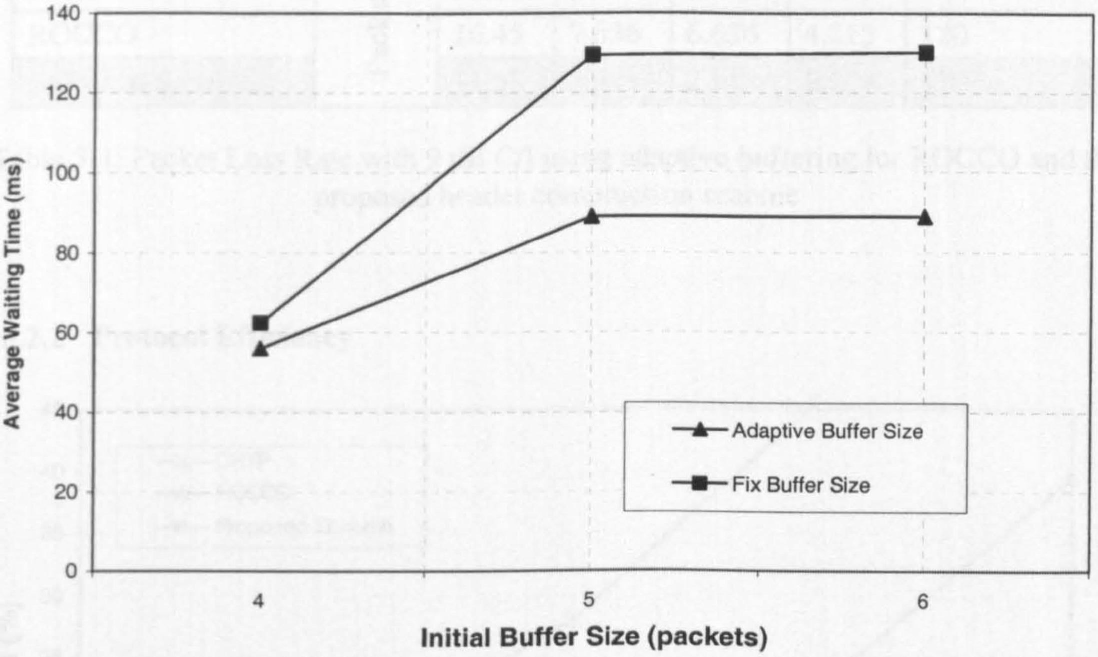


Figure 5-12: Average Waiting Time with Pareto (80,3.5)

Table 5-1 shows the packet loss rate when wireless channel errors are introduced in the simulation. Adaptive Buffering is used both with ROCCO and CRTP, and with the proposed 0-byte header compression scheme. The results indicate that Adaptive Buffering solves only part of the problem for wireless IP applications. Adaptive Buffering does not reduce the header sizes for ROCCO or CRTP, which means that ROCCO and CRTP will always perform worse than a 0-byte header compression scheme when channel errors are present. The last column of Table 5.1 indicates the minimum core network delay condition. It is clear that increasing the core network delay has a negative effect on the packet loss rate performance.

FIX Buffer Size (Packets)		4	5	6	7	Min Delay
CRTP	Packet Loss Rate (%)	30.41	25.548	24.902	23.718	80
ROCCO		12.455	10.818	8.618	3.745	80
Proposed Scheme		7	2.091	1.982	1.467	80
Adaptive Buffer Size		4	5	6	7	
CRTP		34.175	26.485	28.92	27.16	120
ROCCO		16.45	7.636	6.636	4.215	120
Proposed Scheme		11.55	3.818	2.818	2.658	120

Table 5-1: Packet Loss Rate with 9 dB C/I using adaptive buffering for ROCCO and the proposed header compression scheme

5.6.2.2 Protocol Efficiency

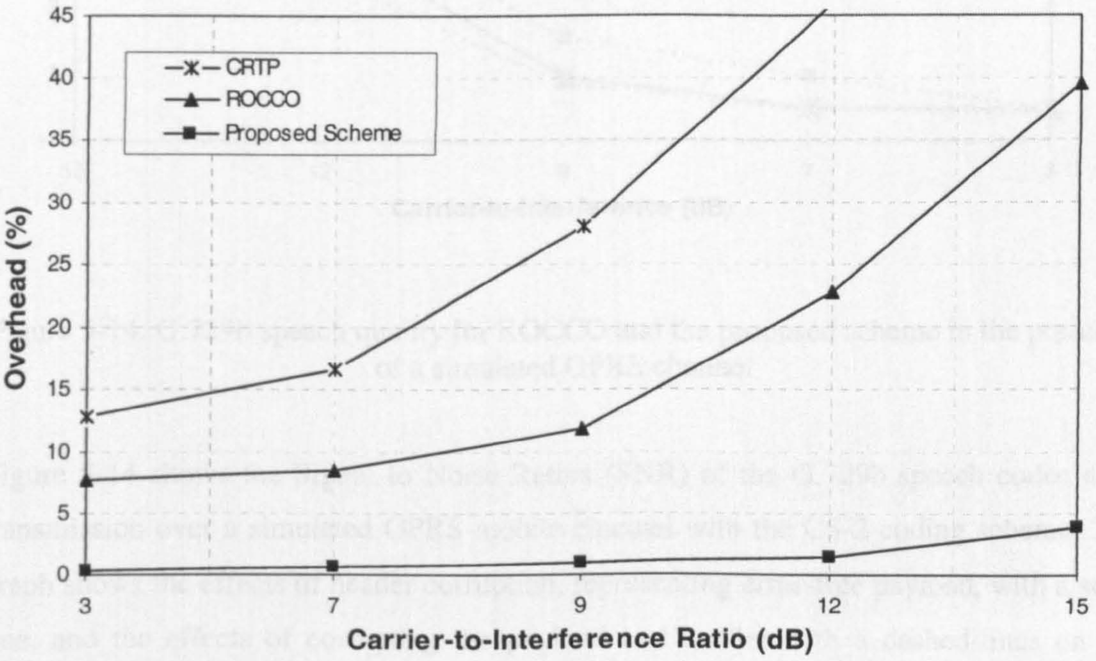


Figure 5-13: Protocol efficiency of ROCCO and the proposed scheme for simulated GPRS transmission

Figure 5-13 shows the results of simulated transmission of speech with ROCCO and with the proposed header compression scheme over GPRS. All header compression schemes use Adaptive Buffering. The Protocol efficiency for the proposed scheme is clearly much better than that for ROCCO, particularly at high bit error rates (low C/I). Although ROCCO reduces the overhead significantly when compared to the full RTP/UDP/IP header, the characteristics of the core network (e.g. overflow and late arrival), combined with wireless channel losses, reduce the compression efficiency of CRTP and ROCCO, meaning that the ideal 2-byte ROCCO header size can rarely be achieved.

5.6.2.3 Speech and Video Codecs Performance

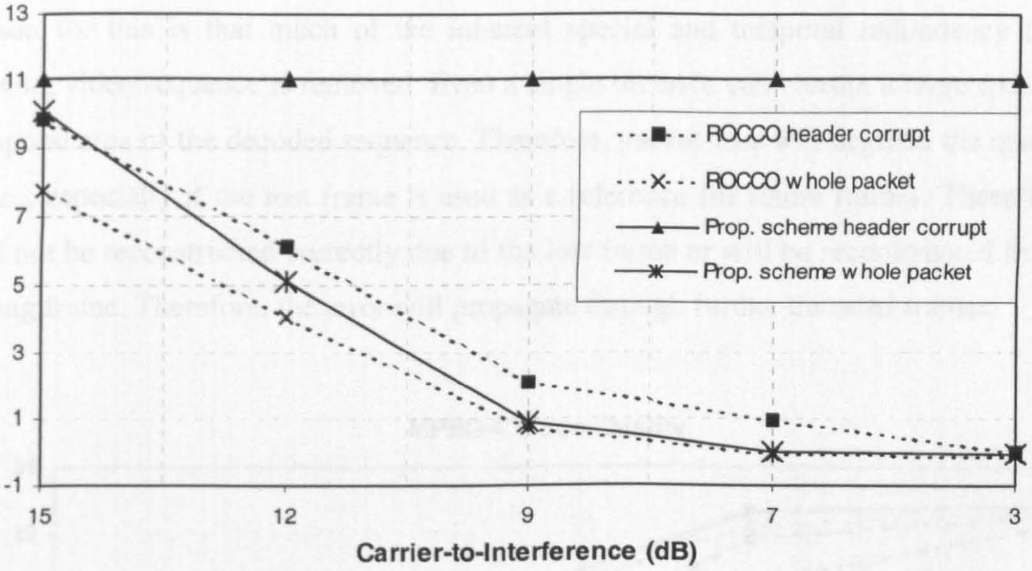


Figure 5-14: G.729b speech quality for ROCCO and the proposed scheme in the presence of a simulated GPRS channel

Figure 5-14 shows the Signal to Noise Ratios (SNR) of the G.729b speech codec after transmission over a simulated GPRS mobile channel with the CS-2 coding scheme. The graph shows the effects of header corruption, representing error-free payload, with a solid line, and the effects of corrupting the payload and header with a dashed-lines on the objective quality. This is shown for both ROCCO and the proposed scheme. Payload damage may cause the frame to be unusable by the speech decoder, depending on the location of the damage.

As the results show, SNR is reduced considerably by the header damage that is caused by lost packets in both cases. The proposed "adaptive-time windowing and adaptive-forward buffering" technique performs significantly better than ROCCO. This is because consecutive packets are not dependent on each other in the proposed scheme. Therefore, unlike ROCCO, packet loss does not affect the following packet. The intelligibility of the speech is improved by using the proposed technique.

Compressed video transmission over mobile channel is more critical than the speech, because the compressed video data is more sensitive to errors. Even a relatively low level of packet loss can have a severe effect on the quality of the decoded video data. The reason for this is that much of the inherent special and temporal redundancy from a moving video sequence is removed. Even a single bit error can corrupt a large spatial and temporal area of the decoded sequence. Therefore, packet-loss will degrade the quality of video, especially if the lost frame is used as a reference for future frames. These frames will not be reconstructed correctly due to the lost frame or will be reconstructed from the wrong frame. Therefore, the error will propagate through further decoded frames.

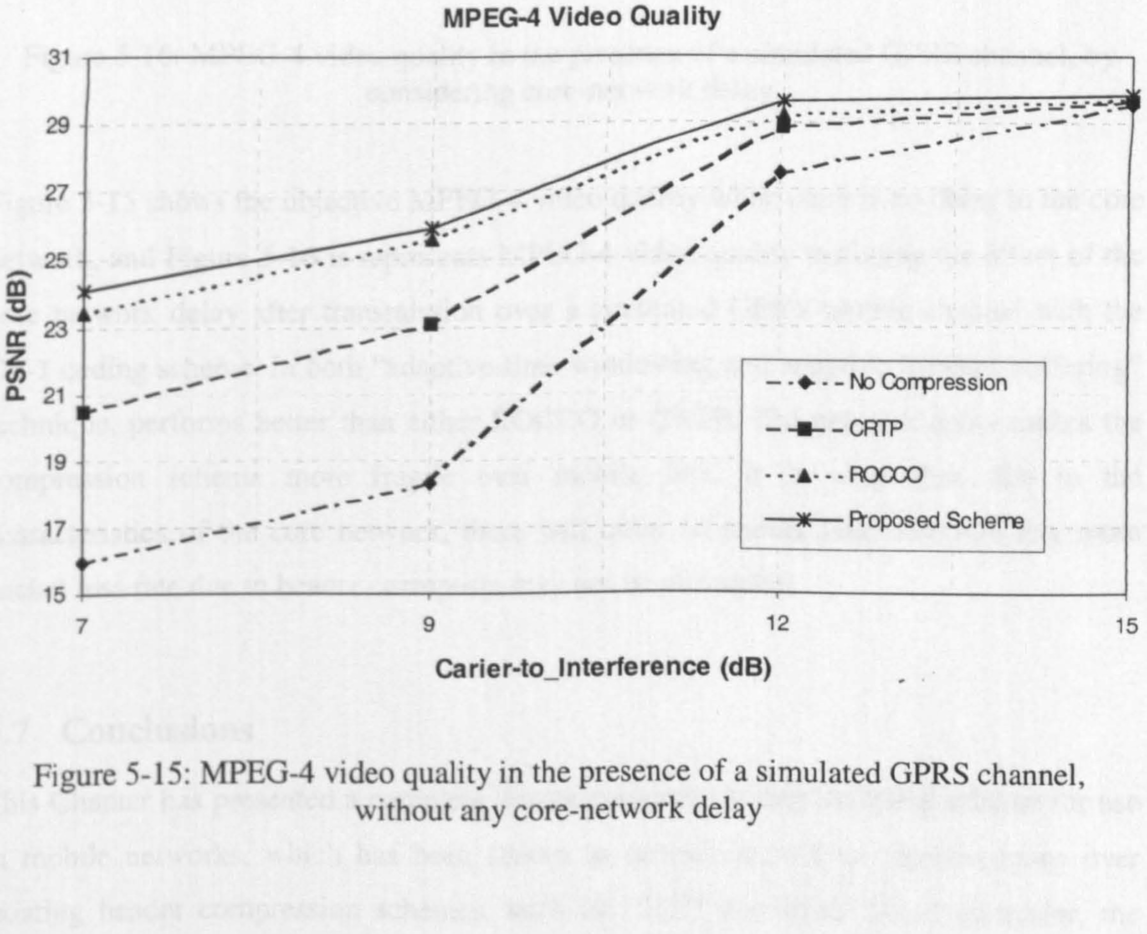


Figure 5-15: MPEG-4 video quality in the presence of a simulated GPRS channel, without any core-network delay

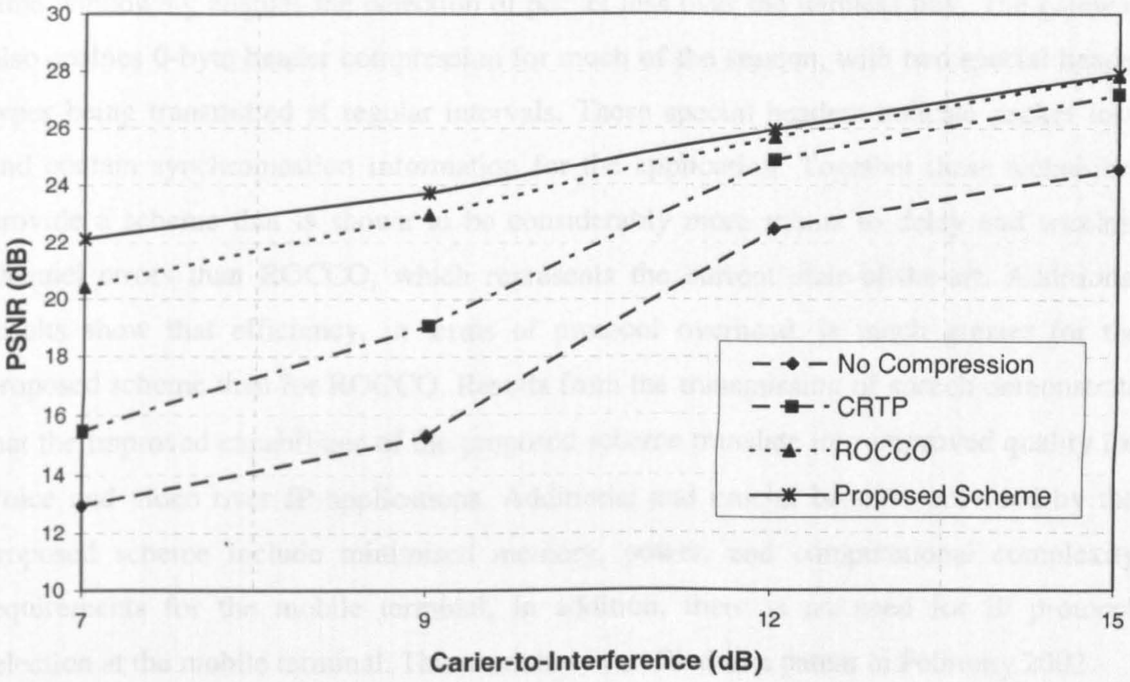


Figure 5-16: MPEG-4 video quality in the presence of a simulated GPRS channel, by considering core-network delay

Figure 5-15 shows the objective MPEG-4 video quality when there is no delay in the core network, and Figure 5-16 is represents MPEG-4 video quality including the effect of the core network delay after transmission over a simulated GPRS mobile channel with the CS-1 coding scheme. In both "adaptive-time windowing and adaptive-forward buffering" technique, performs better than either ROCCO or CRTP. The network delay makes the compression scheme more fragile over mobile link. It is clear that, due to the characteristics of the core network, there will often be packet loss; therefore any more packet loss rate due to header corruption may not be affordable.

5.7 Conclusions

This Chapter has presented a complete header compression and buffering scheme for use in mobile networks, which has been shown to provide significant improvements over existing header compression schemes, such as CRTP and ROCCO. In particular, the existing schemes suffer from low efficiency, high complexity and poor robustness over high bit error rate mobile channels. The proposed scheme uses a gateway at the edge of the core network to smooth out variation in arrival times from the core network. Adaptive time windowing enables the detection of packet loss over the wireless link. The gateway also enables 0-byte header compression for much of the session, with two special header types being transmitted at regular intervals. These special headers indicate packet loss, and contain synchronisation information for the application. Together these techniques provide a scheme that is shown to be considerably more robust to delay and wireless channel errors than ROCCO, which represents the current state-of-the-art. Additional results show that efficiency, in terms of protocol overhead, is much greater for the proposed scheme than for ROCCO. Results from the transmission of speech demonstrate that the improved capabilities of the proposed scheme translate into improved quality for Voice and video over IP applications. Additional and crucial benefits provided by the proposed scheme include minimised memory, power, and computational complexity requirements for the mobile terminal. In addition, there is no need for IP protocol selection at the mobile terminal. This work has been filed as a patent in February 2002.

Chapter 6

6 Conclusions

6.1 Preamble

Multimedia services over wireless environments are likely to be attractive for many users as well as mobile companies. However, they cannot be supported by second-generation mobile networks, which use circuit-switched technology. IP-based mobile networks will grant very high service flexibility and application independence, facilitating a multitude of real-time and interactive services. As a result, the drive towards extending the range of services for mobile users will make packet-switched technology a significant transport technology.

3G networks will enable the provision of a wide range of IP-based multimedia applications, real-time/non-real-time, over wireless links. UMTS is the Third Generation (3G) mobile communication system for much of the world. In UMTS, personal services are based on a combination of fixed and mobile radio services providing a seamless end-to-end service to users by using an IP-based network technology.

Unfortunately, there are a number of problems associated with using packet-switched technology for the delivery of multimedia services over mobile environments. One of the main problems is that wireless channels have limited bandwidth, making radio spectrum the most costly resource in cellular links. Problems occur because, IP-based real-time multimedia services generally require the use of the Real-time Transport Protocols (RTP). It provides end-to-end network transport functions, suitable for transporting real-time application data, such as audio, and video, over multicast and unicast networks. It is typically deployed on top of the UDP/IP protocols. The combined RTP/UDP/IP headers have a length of at least 40 bytes. This includes the IP header (20 octets), the UDP header (8 octets) and the RTP header (12 octets). If IPv6 is used, the total is increased to 60 bytes. When operating over low throughput links, or when transmitting speech or audio streams, which have been compressed to low data rates, the headers are often larger than the payload. Thus, use of RTP results in decreasing transmission efficiency. Originally the RTP/UDP/IP protocols were developed for packet-switched fixed-networks. Significantly large headers are a major issue for multimedia services in mobile environments as bandwidth is expensive.

In light of these facts, two proposed standard header compression algorithms have been implemented and investigated in this research work. In these algorithms, priority is given to the compression efficiency for bandwidth limited mobile channels. The two algorithms can provide efficiency, but not robustness at the same time. For wireless links, the header compression algorithm needs to be error resilient as well as efficient. A good header compression scheme should ensure that the original packet header travels through the air interface without damage that propagates, and also ensures that mobile network resources are exploited in the most efficient manner possible.

6.2 Concluding Overview

In *Chapter 1*, the objectives of the research work are defined. The performance assessment techniques used throughout the thesis have been introduced. Furthermore, the original contributions have also been outlined.

Chapter 2 primarily comprises the background for this thesis. In this chapter, current and future mobile networks are considered. In addition, the required protocols for real-time packet-switched mobile telecommunication are discussed. Briefly, mobile networks will begin to use packet-switched technologies progressively through the introduction of GPRS, EDGE and finally UMTS. At the same time, the QoS of mobile networks will be increased by these technologies. GPRS has significant limitations for multimedia communications. It is likely to have high and varying delay, and low bandwidth, which will probably make it unsuitable for anything other than limited streaming of very low quality multimedia services. EDGE should introduce greater bandwidth, but coverage may be limited due to the sensitivity of the modulation scheme to errors. More sophisticated multimedia applications are likely to have to wait for the introduction of UMTS. Bandwidths greater than 64 kbit/s are likely, while if the specified QoS classes are implemented, then support for real-time multimedia services will be possible, such as video communications.

Also, in this chapter, it is made clear that packet-switched technology will be the core technology for the provision of these services. In particular, the role of IP protocol stacks, their flexibility advantages, and their impact on mobile communications is discussed. Different protocol stacks, which are used for various purposes, are briefly outlined, and the real time protocols, which are needed for real-time services and are also used in research work, are highlighted and explained in detail. Moreover two major applications, Speech and Video applications, including different speech codecs, G.723.1, G.729b, AMR, and a video codec, MPEG-4, are discussed and important points for packet-switched mobile networks are highlighted. In addition, the important issues that need to be considered for packet-switched technology in mobile channels are mentioned.

Chapter 3 is the first chapter that includes author contributions. At the beginning of this chapter, the real-time protocol stacks, RTP/UDP/IP, are examined. The problems of using these protocols without compressing them and the advantages of header compression were listed and discussed. A header capture and analyzer program is implemented to examine the characteristics of the header fields in detail, and the way that they change during multimedia transmission. After that, the first proposed real-time header compression algorithm by the IETF standard committee, CRTP, was fully implemented. The performance of this scheme was evaluated over a GPRS mobile channel, and the results are presented including the ideal case outcome of CRTP. The packet loss rate results show that CRTP does not perform well over mobile channels. The main problem of these poor performances is that the error on one packet's compressed header propagates and many following packets get discarded, until a full header arrives at the de-compressor. Two previously proposed error resilience methods, TWICE [1] and periodic refreshes [1] were implemented.

The TWICE scheme is a de-compressor internal mechanism, intended to mitigate the effects of any packet loss. It requires the UDP checksum to be enabled to verify its repair attempts, which costs two extra octets for every packet. It increased the packet loss rate performance slightly but not sufficiently.

Periodic Refresh is proposed for simplex links, where the feedback channel is not available, and for links that have high delay. Periodic refresh methods update the context at fixed intervals. It does not require any context update requests from the de-compressor. It decreases the packet loss rate, if it updates the compression context faster than the full header request based scheme. The average header size does not vary according to the channel conditions, which makes it unreliable over mobile channels, even though it does not require a feedback channel like TWICE.

In this chapter, the REFERENCE_HEADER packet format is proposed to limit error propagation. When this packet format is used, the header fields are compressed based on the last Reference_Header packet information that has been sent. This way any error on the compressed header does not propagate, as long as the Reference_Header packet arrives at the de-compressor correctly. It increases the performance in terms of the packet loss rate over GPRS mobile channels; however, compression efficiency is considerably reduced. As an alternative solution, a slow-start update scheme (SSUS) is proposed by the author. This method is employed at the encoder. It is based on feedback of the channel conditions, including end-to-end delay. The encoder decides when full headers need to be sent. The performance of the packet loss rate results show that the SSUS method improves the packet loss rate performance compared to standard CRTP. However, it increases the average header size significantly. It is the most costly method regarding bandwidth.

It can be summarized that the performance of CRTP over lossy links, and especially channels with long roundtrip times was not sufficient. With all the error resilience methods that have been tried, the performance of the CRTP is still worse than the Ideal case in terms of the packet loss rate and the efficiency.

Chapter 4 presents an alternative header compression scheme, ROCCO. ROCCO, as CRTP, was proposed by the IETF committee in 1999 and is still under going research.

ROCCO has been introduced as a 'robust and efficient' header compression scheme in cellular environments. It has been implemented and evaluated over GPRS mobile channels as for CRTP. The results show that, it is much more robust than CRTP, but in terms of efficiency, there is not a great improvement over very lossy mobile channels (e.g. 7 dB CS-2 GPRS channel). Nevertheless, the results show that ROCCO provides much greater robustness, which improves speech and video quality, and provides better compression efficiency than CRTP. In this chapter, two error resilience methods are introduced by the author: Adaptive_Reference Update Scheme (ARUS) and prioritisation scheme, and a new compressed packet format, the Adaptive_Reference header. These methods are evaluated and compared with standard ROCCO. The results show that, the two methods help improve robustness and decrease the packet lost rate. However, efficiency is reduced, and the average header size increases under all conditions. In terms of computational complexity, ROCCO is much worse than CRTP. Considering header compression will be run on a battery-operated handset, implementation will be another challenge for researchers.

To summarise Chapters 3 and 4, the existing header compression schemes, CRTP and ROCCO, even with error resilience methods, suffer from low efficiency, high complexity and poor robustness over high bit error mobile environments, especially for continuous time varying channels.

In light of these facts, *chapter 5* presents a novel system, "Adaptive time-windowing and adaptive forward-buffering scheme", which minimises header sizes, conserving bandwidth and radio spectrum, and also minimises the effects of varying transmission delay. Within this scheme, an "application-defined packet" and a "smart packet" are proposed, which improve robustness and efficiency.

At the beginning of this chapter, the main reasons for the poor performance of the CRTP and ROCCO algorithms are highlighted. In summary, these techniques use lossless compression algorithms, which mean that the result of the de-compression must be bit-by-bit identical with the original compressed header. Therefore, the results show that for certain real-time applications and characteristics of radio links, the presence of even a single octet of header may result in a significant decrease in spectrum efficiency compared to existing circuit-switched technology. An other reason for poor performance is, in best-effort packet transmission scenarios, such as provided by IP networks, data packets are subjected to time-varying delay as a result of differing levels of congestion and varying loads in different parts of the network. Variations in packet delay also known as jitter cause real-time packets to arrive at their destination in uneven patterns. This often results in packets either arriving out of order, or alternatively with a high degree of variability in packet arrival time. Typically, the solution to jitter problems is, mobile terminals must be able to buffer the input streams to smooth out the variability, which will require significant amounts of memory to store the packets. In the case of multi-stream transmission, stream synchronisation must also be performed. Even then, there is no guaranteed service, since mobile environments can change very frequently, and cannot be predicted in advance. Also, any unexpected error on the header will cause extra processing time delay at the de-compressor, which can be very serious. The higher degree of variability introduced by the radio access network reduces the efficiency of ROCCO significantly, even though it provides reasonable robustness. Therefore, the average required overhead can never be one byte due to the following:

- The time-varying characteristics of mobile links
- The required overheads during the initialisation process
- The Core Network transmission delay or the variability of packet arrival times

The proposed scheme makes use of the information related to a bit-stream that is supported by the standard applications (e.g. types of video codec or speech codec), to overcome the above problems. It is designed to provide smooth play-out of packets over the air interface on the downlink, and minimises the computational complexity and power requirements on the handset by providing 0-byte header compression. The main advantage of this scheme is that consecutive packets are independent of each other over wireless links, which cannot be achieved using CRTP or ROCCO. Therefore, there is no risk of consecutive packets being discarded due to corrupted header information. One of the essential requirements of this scheme is to place an Edge Gateway (Proxy) at the Edge of the Core Network (EoCN) between the end-user terminal and the CN. This proxy has different functionalities for uplink and downlink transmission, which are explained in detail in the Chapter 5.

The special headers introduced in this work indicate packet loss, and contain synchronisation information for the application. Together these techniques provide a scheme that is shown to be considerably more robust to delay and wireless channel errors than ROCCO. Additional results show that efficiency, in terms of protocol overhead, is much greater for the proposed scheme than for ROCCO. Results from the transmission of speech demonstrate that the improved capabilities of the proposed scheme translate into improved quality for Voice and video over IP applications. Additional and crucial benefits to the use of the proposed scheme include minimised memory, power, and computational complexity requirements for the mobile terminal and also remove the need for IP protocol selection at the mobile terminal. This work has been filed as a UK patent on 14th February 2002, and the patent application number is 0203511.1.

6.3 Future Work

"Header compression" or "minimised header" concepts in wireless environments will remain major concerns, especially for commercial industries. And always will be a hot topic many more years in research areas. This is because, the cost of wireless spectrum means that efficiency in the radio link is always going to be a prime concern when designing cellular radio access networks.

The possibilities for further development and research in this area can be split into two parts:

- Enhance the existing proposals
- Try different approaches for research

The first part considers the development of the existing proposals, included in this thesis, and produces more suggestions to improve the robustness as well as the efficiency over mobile channels.

6.3.1 Enhance the existing proposal

In this thesis, the packet loss rate performance improvement is given priority for the CRTP and ROCCO header compression schemes. All existing error resilience and other proposed techniques focus on robust compressed header transmission over mobile channels. Although these techniques improve the packet loss rate performance, they require more bandwidth, which reduces the efficiency of the compression scheme. More research can be performed to improve the efficiency.

In addition, the proposed scheme needs to be tested for streaming multimedia services as well as for non-real-time packet communications, such as TCP, FTP, and HTTP. For these applications, since they are not very delay sensitive, effective feedback channels can be investigated for both end air interfaces, between terminal-to-Edge Proxy on the uplink, and Edge Proxy-to-terminal on the downlink.

The following areas can be investigated for both CRTP and ROCCO header compression schemes;

- **Implement Integrated Error Protection Mechanisms using block codes**

Error protection mechanisms can be implemented to protect the compressed header in very error-prone environments. By considering header compression based on packet-switched networks, a block code on the compressed header can be implemented. The reason for the use of a block code instead of a convolutional code is the low number of bits in compressed headers.

- ***Investigate Unequal Error Detection (UED) and Unequal Error protection (UEP)***

The compressed header dataset can be divided onto three categories according their sensitivity. The dataset that can damage the compression context should be given the highest priority. This way, an unequal error protection scheme can be employed to protect the most sensitive dataset best, the second sensitive dataset less, and finally the third sensitive bits least.

- ***Investigate more radical Prioritisation***

Most of the speech codecs uses VAD (Voice Activity Detector), and send active speech and silence periods separately. The active frame is more important than the silence therefore the prioritisation scheme can be used to make header compression schemes adaptive.

For Video transmission, video frames can be split into two; motion information and texture information. In this way the information can be sent in different packets. Motion information is much more important than texture information, therefore the prioritisation scheme can be used to make the header compression scheme adaptive and more robust for the motion packet.

- ***Enlarge the compression scheme scope***

Investigate making a multi-functionality-robust header compression scheme that is independent of call type, (i.e. HTTP, FTP, TCP and RTP), for mobile-to-mobile, mobile-landline, landline-mobile channels.

6.3.2 Different approaches for research

Current research focuses on the header compression algorithm, which is a straightforward solution for reducing significantly large protocol stacks in mobile environments, while maintaining the interoperability between different networks. However, as described and tackled in this thesis, it is urged that the current protocol arrangements employed in mobile communication systems such as UMTS are inefficient, susceptible to errors and ill-suited to the task of transporting time-sensitive multimedia information. Therefore, this section proposes a more radical solution than the header compression attempts, which is:

> *"To create a new transmission link protocol that enables the IP based stack to be replaced by a bandwidth efficient and error robust scheme for multimedia applications"*

This proposed scheme aims to efficiently replace several layers of the protocol stack and allow for a single universal protocol to be employed over mobile links. It can be referred to as the ***Multimedia Radio Link Protocol (MRLP)***. In order to ensure compatibility with the current network protocols, the MRLP must be transparent to devices on other networks. Because of that, the research will include the design of the *Universal network Adaptation Layer (UNAL)* for a wide range of multimedia services including graphics, synthetic video, virtual reality (3D) scenes, natural video, multi-channel and single channel audio, speech and many other as yet unforeseen services. A *universal multimedia container (UMC)* will be provided at the UNAL interface for packing the media service and for specifying the required Service Grade (SG) requirements. UNAL will provide functionality for the conversion of all incoming IP media packets into the MRLP format for the downlink, with a minimum of overhead. Conversely, on the uplink media containers transmitted from the mobile terminal to the network must be encapsulated within RTP packets before transmission to the core network.

In summary, the proposed MRLP aims to provide a single lightweight low-overhead protocol that spans several traditional layers, while remaining at the same time robust to channel errors.

Apendix A

List of Publications

- A. Cellatoglu, S. Fabri, A.M. Kondoz, "Use of Prioritised Objected Video Coding for the Provision of Multiparty Video Communications in Error Prone Environments", IEEE Vehicular Technology Conference, VTC, Amsterdam, Netherland, October 99.

- S. Fabri, A. Cellatoglu, S. Worrall, A.M. Kondoz, "Transmission of Multimedia Services over GPRS using MPEG-4 Coded Video", IEEE Vehicular Technology Conference, VTC, Amsterdam, Netherland, October 99.

- A. Cellatoglu, S.T. Worrall, S.N. Fabri, A.H. Sadka, A.M. Kondoz "Performance of RTP/UDP/IP header compression in cellular networks", London Communications Symposium, London, UK, September, 2000

- A. Cellatoglu, S. Fabri, S. Worrall, A.H. Sadka, A.M. Kondoz, "Robust Header Compression for Real-Time Services in Cellular Networks", IEE 3G 2001, London, March 2001, pp. 124-128.

- S. Dogan, A. Cellatoglu, A. H. Sadka, and A. M. Kondoz, "Error-resilient video transcoding for 3G internetwork communications", IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Wireless Video, September, 2002.

- S. Dogan, A. Cellatoglu, A. H. Sadka, and A. M. Kondoz, "Error-resilient MPEG-4 video transcoder for bit rate regulation", Procedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics, SCI'2001, Vol.XII, Part II, pp.312-317, Orlando, FL, USA, 22-25 July 2001.

- A. Cellatoglu, S. Fabri, S. Worrall, A.M. Kondoz, "Robust and Efficient Multimedia Packet Transmission Technique over Wireless Networks", Signal Processing Image Communication Magazine (submitted)

- A. Cellatoglu, S. Fabri, A.M. Kondoz, "Header Compression for Mobile Access Networks", UK Patent Application No: GB0203511.1

Apendix B

List of Abbreviations

3G	Third Generation
8-PSK	8-Phase Shift Keying
ACK	Acknowledgment
ADP	Application Defined Packet
AMR	Adaptive Multi-Rate
ARUS	Adaptive Reference Update Scheme
BER	Bit-error-rate
BS	Base Station
BW	Bandwidth
C/I	Carrier-to-interference Ratio
CCSR	Centre for Communication Systems Research
CELP	Code-Excited Linear Prediction
CN	Core Network
CRC	Cyclic Redundancy Check
CS	Coding Scheme
CRTP	Compression RTP/UDP/IP
CSRC	Contributing Source
DLeP	Down-Link Edge Proxy
EDGE	Enhanced Data Rates for GSM Evaluation
EFR	Enhanced Full Rate
EGPRS	Enhanced GPRS

EoCN	Edge Of the core Network
ETSI	European Telecommunication Standards Institute
FIFO	First-in First-out
FRE	Frame Error Rate
GERAN	GSM/EDGE Radio Access Network
GMSK	Gaussian Minimum Shift Keying
GPRS	General Packet Radio Service
GSM	Global System for Mobile telecommunications
GW	Gateway
HC	Header Compression
I	Intra
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISDN	Integrated Service Digital Network
ISO	International Standards Organisation
ITU	International Telecommunications Union
MAC	Medium Access Control
MPEG	Motion Picture Experts Group
MSC	Mobile Switching Centre
NACK	non-acknowledgment
P	Inter/predictive
PDCP	Packet Date Converge Protocol
PDTCH	Packet Data Traffic Channel
PDU	Packet Data Unit
PLR	Packet Loss Rate
PSNR	Peak-to-peak Signal-to-Noise Ratio
PSTN	Public Switched Telephone Network
PT	Packet Type
QoS	Quality of Service
RAB	Radio Access Bearer
RAN	Radio Area Network
R-DL	Radio- Down Link
RLC	Radio Link Control
RNS	Radio Network Subsystem

ROHC	Robust Header Compression
RP	Reverse Proxy
RTSP	Real-time Streaming Protocol
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
R-UL	Radio Up_Link
SDP	Session Description Protocol
SN	Sequence Number
SIP	Session Initiation Protocol
SNR	Signal-to-Noise Ratio
SP	Smart Packet
SSUS	Slow Start Update Scheme
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
ToS	Type of Service
TS	Timestamp
TTL	Time To Live
TU	Typical Urban
UDP	User Data Protocol
UEP	Unequal Error Protection
ULeP	Up-Link Edge Proxy
UMTS	Universal Mobile Telecommunication Service
UTRAN	UMTS Terrestrial Radio Access Network
VoIP	Voice over IP

Bibliography

[1] Casner S., and Jacobson V., "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", *RFC 2508*, February 1999.

[2] Degermark M., Nordgren B., and Pink S., "IP Header Compression", *RFC 2507*, February 1999.

[3] ETSI/SMG, GSM 03.64 1998 "Overall Description of the GPRS Radio Interface Stage 2", *V. 5.2.0.*, 1998.

[4] Brasche G., and Walke B., "Concepts, Services and Protocols of the New GSM Phase 2+ General Packet Radio Service", *IEEE Communications Magazine*, pp.94-104, October 1997.

[5] SMG. "GSM 05.03 v6.1: Digital cellular telecommunications system (GSM Radio Access Phase 3); Channel Coding", January 1998.

[6] Jonsson L., Degermark M., Hannu H., and Svanbro K., "Robust Checksum-based header Compression (ROCCO)", *IETF Internet Draft*, January 2000.

[7] Jacobson V. "Compressing TCP/IP Headers for Low-Speed Serial Links", *Internet-Draft RFC 1144*, February 1990.

[8] Postel Jon, "Internet Protocolv4 Specification", *RFC 791*, September 1981

[9] ETSI/SMG, "Study of H.323 as a multimedia protocol for a GPRS/UMTS real time voice and video service" Tdoc SMG mm 015/98, V. 1.1, September 1998.

[10] Postel J., "The UDP Protocol", *Internet-Draft, RFC 768*, August 1980.

[11] Deering S., and Hinden R., "Internet Protocol, Version 6 (IPv6) Specification", ", *Internet-Draft RFC 2460*, December 1998.

[12] Schulzrinne S., Casner S., Frederic R., and Jacobson V., "RTP: A transport protocol for real-time applications", RFC 1889, January 1996.

[13] Westberg L., and Lindqvist M., "Realtime Traffic over Cellular Access Networks", Internet-Draft, May 22, 2000. <draft-westberg-realtime-cellular-02.txt>

[14] Kostas T.J., Borella M.S., Sidhu I., Schuster G.M., Grabiec J., and Mahler J., "Real-time Voice Over Packet-Switced Networks", IEEE Network magazine, Jan/Feb 1998.

[15] ElGebaly H., "Reactive Mechanisms for Recovering Audio Performance in Multimedia Conferencing Over Packet Switched Networks", *Intel Technology Journal Q3,* 1999.

[16] ETSI/SMG, "The Role of the Internet in UMTS", *Tdoc SMG12,* June 1998.

[17] Ahuja R.S., and Murti G.K., "Packet Telephony", *Bell Labs Technical Journal,* spring 1997.

[18] ElGebaly H., "Characteristic of Multimedia Streams of an H.323 Terminal", Intel Technology Journal Q2, 1998.

[19] Kikuchi Y., Nomura T., Fukunaga S., Matsui Y., and Kimata H., "RTP payload format for MPEG-4 Audio/Visual streams" Internet Daraft, draft-ieft-avt-rtp-mpeg4-es-00.txt.

[20] Cai J., and Goodman J.D., "General Packet Radio Service in GSM", IEEE Communications Magazine, October 1997.

[21] Fabri S.N., Kondoz A., Tatesh S., and Demetrescu C., "Proposed evolution of GPRS for the support of voice services", IEE Proc., Commun., Vol. 146, No. 5, October 1999.

[22] Dowden D.C., Gitlin R.D., and Martin R.L., "Next-Generation Networks", *Bell Labs Technical Journal,* October-December 1998.

[23] Civanlar M.R., "Protocols for Real-time Multimedia Data Transmission Over the Internet", IEEE Magazine, 1998.

[24] Bolot J.C., "Characterizing End-to-End Packet Delay and Loss in the Internet", *In Journal of High-Speed Networks, vol. 2, no. 3, pp.* 305-323, December 1993.

[25] ETSI, "UMTS;Service aspects Mobile multimedia services including mobile Intranet and Internet services", TR 22.60, V3.0.0, April 1998.

[26] ETSI/SMG, "UMTS;Real Time Multimedia in UMTS", *TR MM.mm V0.0.0,* February 1998.

[27] Chikarmane V., Williamson C.L., Bunt R.B., and Mackrell W.L., "Multicast Support for Mobile hosts using Mobile IP:Design issues and proposed architecture", *Mobile Networks and Applications* 3, pp. 365-379, 1998.

[28] Nilson T., "Toword third-generation mobile multimedia communication", *Ericsson Review No. 3,* 1999.

[29] Kumar V., "Supplementary Service in the H.323 IP Multimedia telephony Networks", IEEE Communications Magazine, July 1999.

[30] Thom G.A., "H.323: The Multimedia Communications Standart for Local Area Networks", *IEEE Communications Magazine*, December 1996.

[31] August K.G., Lawrence, and Saltzeberg B.R., "An Introduction to Future Communications Service and Access", *Bell Labs Technical Journal*, April-June 1999.

[32] G. AP Eriksson, B. Olin, K. Svanbro, D. Turina, "The Challenges of Voice-Over-IP-Over-Wireless", Ericsson Review, No. 1,

[33] 3GPP, "Architecture for an All IP network", *3G TR 23.922 V1.0.0*, October 1999.

[34] ITU-T, Recommendation H.323, "Terminal for Low BitOrate Multimedia Communication over Non-Guaranteed Bandwidth Networks", 1996.

[35] ITU-T, Recommendation G.723.1, "Dual Rate Speech Coder For Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s", March 1996.

[36] ITU-T, Recommendation G.729, "Coding of speech at 8 kbit/s using conjugate structure algebraic-code-excited linear-prediction (CS-ACELP)", November 1996.

[37] Hersent O., Gurle D., and Petit J.P., "IP Telephony; Packet-based multimedia communications systems", *Addison Wesley* 2000.

[38] ETSI, Digital cellular telecommunications sytems (Phase 2+), "Adaptive Multi-Rate (AMR) speech transcoding", *EN 301 704 V7.2.0*, December 1999.

[39] Rappaport T.S., "Wireless Communications", Principles & Practice, 1996.

[40] Bull D., Canagarajah N., and Nix A., "Mobile Multimedia Communications", *Signal Processing and Its Applications, Academic Press*, 1999.

[41] Cox R.V, "Tree New Speech Coders from the ITU Cover a Range of Applications", IEEE Communications Magazine, September 1997.

[42] Ekudden E., Haden r., and Svedberg J., "The Adaptive Multi-Rate Coder", *IEEE Speech Workshop*, US, 1999.W

[43] Wong W.T.K., Mack R.M., Cheetham B.M.G., Sun X.Q., "Low Rate Speech coding for telecommunications", *BT Technical Journal Vol. 14, No. 1*, January 1996.

[44] Wong W.T.K., Mack R.M., Cheetham B.M.G., Sun X.Q., "Low Rate Speech coding for telecommunications", *BT Technical Journal Vol. 14, No. 1*, January 1996.

[45] S. J. Perkins, M. W. Mutka, "Dependency Removal for Transport Protocol Header Compression over Noisy Channels",

[46] M. Engan, S. Casner, C. Bormann, "IP Header Compression over PPP", *IETF RFC-2509*, February 1999,

[47] J. Border, M. Kojo, J. Griner, G. Montenegro, Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", *IETF-draft*, June 2001,

[48] S. Dogan, A. Cellatoglu, A. Sadka, A. Kondoz, "Error-Resilient MPEG-4 Video Transcoder for Bit Rate Regulation Purposes in Heterogeneous Multimedia Networking", *Proceedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics*, SCI'2001, Vol.XII, Part II, pp.312-317, Orlando, FL, USA, 22-25 July 2001.

[49] G. Fodor, F. Persson, B. Williams, "Proposal on New Service Parameters (Wireless Hints) in the Controlled Load Integrated Service", *IETF-draft*, January 2001,

[50] P. Reynolds, "Mobile Internet Service Provider (MISP) Requirements for a Wireless Internet Franewor (WIF), May 2001,

[51] B. Thompson, T. Koren, D. Wing, "Tunneling Multiplexed Compresses RTP ("TCRTP"), IETF-draft, July 19, 2001,

[52] V. Varsa, M. Karczewicz, G. Roth, R. Sjoberg, T. Stockhammer, G. Liebl, "Common Test Conditions for RTP/IP over 3GPP/3GPP2", ITU-T, 1st May 2001,

[53] M. Handley, J. Crowcroft, C. Bormann, J. Ott, "The Internet Multimedia Conferencing Architecture", *IETF-draft*, April 2000.

[54] T. F. La Porta, L. Salgarelli, G. T. Foster, "Mobile IP and Wide Area Wireless Data" *Proceedings of IEEE Wireless Communications and Networking Conference 199*, pp. 21-24, September 1999,

[55] 3GPP, "Technical Specification, 3rd generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS); Service description, Stage 1 (Release 2000)", 3G TS 22.06 V4.2.0, January 2001

[56] S. M. Bellovin, J. Ioannidis, A. D. Keromytis, R.R. Stewart, "On the Use of SCTP with IPsec", IETF-draft, 2001.

[57] Q. Zhang, W. Zhu, Y. Q. Zhang, "QoS-Adaptive Multimedia Streaming over 3G Wireless Channels", Second International Sumposium on Mobile Multimedia Systems and Applications (MMSA) 2000, Nov., 2000, Delft, The Netherlands.

[58] J. Chan, A. Seneviratne, B. Landfeldt, "The Challenges of Provisioning Real-Time Services in Wireless Internet", Telecommunications Journal of Australia, Vol.50, No.3, pp. 37-48, Spring 2000.

[59] R. G. Tebbs, "Real-Time IP Facsimile: Protocol and Gateway Requirements", *Bell Labs Technical Journal,* April-June 1999.

[60] H. Schulzrinne, J. Rosenberg, "Internet Telephony: Architecture and Protocols an IETF Perspective", Proceeding IEEE Network Magazine May/June 1999.

[61] M. Degermark, H. Hannu, L. E. Jonsson, K. Svanbro, "CTRP over Cellular Radio Links", December 10, 1999.

[62] D. Partain, G. Karagiannis, P. Wallentin, L. Westberg, "Resource Reservation Issues in Cellular Radio Access Networks", IETF-draft, June 2001,

[63] A. H. Li, "An RTP Payload Format for EVRC, SMV and Other Frame-Based Vocoders", IETF-draft, November 2001.

[64] S. Casner, V. Jacobson, T. Koren, B. Thompson, D. Wing, P. Ruddy, A. Tweedly, J. Geevarghese, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", November 2000,

[65] Z. Liu, K. Le, K. C. Leung, C. Clanton, "Scalable, Robust, Efficient Dictionary-Based Compression (SCRIBE)", IETF-draft, July 2001,

[66] J. Harris, "The Future of Radio Access in 3G", *BT Technol J,* Vol.19, No.1, January 2001.

[67] M. D. Cookson, D. G. Smith, "3G service Control", *BT Technol J,* Vol.19, No. 1, January 2001,

[68] J. Chen, K. J. R. Liu, "Joint Source-channel Multistream Coding and Optical Network Adapter Design for Video Over IP", *IEEE Transactions on Multimedia,* Vol. 4, No. 1, March 2002,

[69] H. Hameleers, and C. Johansson, "IP Technology in WCDMA/GSM Core Networks", *Ericsson Review,* No.1, 2002,

[70] S. R. Ahuja, K. G. Murti, "Packet Telephony", *Bell Labs Technical Journal,* Spring 1997,

[71] J. Wang, P. J. McCann, P. B. Gorrepati, C. Z. Liu, "Wireless Voice-over-IP and Implications for Third-Generation Network Design", *Bell Labs Technical Journal,* July-September 1998,

[72] T. Murphy, "The CDMA2000 Packet Core Network", *Ericsson Review,* No.2, 2001,

[73] F. Muller, J. Sorelius, D. Turina, "Further Evolution of the GSM/EDGE Radio Access Network", *Ericsson Review,* NO. 3, 2001,

[74] F. Yang, Q. Zhang, W. Zhu, Y. Q. Zhang, " An Efficient Transport Scheme for Multimedia Over Wireless Internet", *3G Wireless 01,USA,* June 2001,

[75] K. Fujimoto, S. Ata, M. Murata, "Statistical Analysis of Packet Delays in the Internet and Its Application to Playout Control for Streaming Applications", *IEICE Trans. On Communications,* Vol. E00-B, No. 6, June 2001,

[76] Kaloxylos, G. Papageorgiou, P. Papageorge, L. Merakos, "Smart Buffering Technique for Lossless Hard Handover in Wireless ATM Networks",

[77] D. Covarrubias, C. Merla, "Optimizing the Buffer Size for Packet Voice Transportation Through Fast Packet Switching Networks",

[78] R. Ludwig, A. Konrad, A. D. Joseph, R. H. Katz, "Optimizing the End-to-End Performance of Reliable Flows over Wireless Links", *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking* August 1999.

[79] H. Adiseshu, G. Parulkar, G. Varghese, "A Reliable and Scalable Striping Protocol", *ACM SIGCOMM, Vol. 26, No. 4,* October 1996.

[80] Q. Zhang, Y. Q. Zhang, W. Zhu, "Resource Allocation for Audio and Video Streaming over the Internet", *ISCAS 2000 – IEEE International Symposium on Circuits and Systems,Geneva, Switzerland,* May 28-31, 2000,

[81] M. Degermark, H. Hannu, L. E. Jonsson, K. Svanbro, " Evaluation of CRTP Performance over Cellualr Radio Links", *IEEE Personal Communications,* August 2000.

[82] Giovanardi, G. Mazzini, M. Zorzi, "Improved Header Compression for TCP/ IP over Wireless Links", *Electronics Letters,* Vol.36, No. 23, 9th November 2000,

[83] J. C. R. Bennett, C. Partridge, N. Schectman, " Packet Reordering is Not Pathological Network Behavior", *IEEE/ACM Transactions on Nteworking,* Vol. 7, No. 6, December 1999,

[84] Q. Zhang, W. Zhu, G. Wang, Y. Q. Zhang, "Resource Allocation with Adaptive QoS for Multimedia Transmission over W-CDMA Channels", *Conf. Image Processing (ICIP) 2000,* Vancouver, September, 2000

[85] J. Parantainen, S. Hamiti, "Delay Analysis for IP Speech over GPRS", IEEE VTC 1999.

[86] Furuskar, P. de Bruin, A. Simonsson, "Controlling QoS for Mixed Voice and Data Services in GERAN- the GSM/EDGE Radio Access Network",

[87] M. Eriksson, A. Furuskar, M. Johansson, "The GSM/EDGE Radio Access Network- GERAN; System Overview and Performance Evaluation", *The IEEE Annual Vehicular Technology Conference*, VTC2000-spring, Tokyo-Japan, May 2000.

[88] M. Eriksson, A. Furuskar, F. Kronestedt, C. Lindheimer, S. Mazur, J. Molno, C. Tidestav, "System Overview and Performance Evaluation of GERAN- The GSM/EDGE Radio Access Network",

[89] K. V. D. Wal, M. Mandjes, H. Bastiaansen, "Delay Performance Analysis of the New Internet Services with Guaranteed QoS", IEEE Proceeding Vol.85, No.12, December 1997.

[90] V. Brazauskas, R. Serfling, "Robust and Efficient Estimation of the Tail Index of a Single-Parameter Pareto Distribution", *North American Actuarial Journal Final Revision,* February 2000,

[91] V. Brazauskas, R. Serfling, "Robust and Efficient Estimation of the Tail Index of a Single-Parameter Pareto Distribution", *North American Actuarial Journal Final Revision,* February 2000,

[92] C. Borman, "Providing integrated services over low-bitrate links", *IETF Draft* 1999.

[93] E. Gustafsson, A. Jonsson, E. Hubbard, J. Malmkvist and A. Roos, "Requirements on Mobile IP from a Cellular Perspective", *Internet-draft*, june 1999.

[94] M. Allman, and D. Glover, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", *IETF request,* January 1999.

[95] H. Hannu, k. Svanbro, and L. E. Jonsson, "ROCCO Performance Evaluation", *IETF-draft*, November 2000.

[96] K. Svanbro, "Lower Layer Guidelines for Robust RTP/UDP/IP Header Compression", *IETF-draft*, November 2001.

[97] J. Sjoberg, M. Westerlund, A. Lakaniemi, and Q. Xia "RTP payload format and file storage format for the Adaptive Multi-rate (AMR) and Adaptive Multi-rate Wideband (AMR-WB) audio codecs", *IETF-draft*, February 2002.

[98] K. Svanbro, H. Hannu, L-E. Jonsson and M. Degermark, "Wireless Real-time IP Services Enabled by Header Compression", *IEEE VTC'00*, 2000.

[99] A. Martensson, T. Einarsson, and L-E Jonsson, "ROCCO Conversational Video Profiles", *IETF-draft*, September 2000.

[100] M. C. Bale, "Voice and Internet multimedia in UMTS networks", *BT Technical Journal, Vol., 19, No., 1*, January 2001.

[101] A. Miyazaki, H. Fukushima, T. Wiebke, R. Hakenberg, and C. Burmeister, "Robust Header Compression using Keyword-Packets", *IETF-draft*, November 2000.

[102] R. Koodli, C. E. Perkins, and N. Tiwari, "Context Relocation for Seamless Header Compression in IP Networks", *IETF-draft*, July 2001.

[103] T. Koren, S. Casner, J. Geevarghese, B. Thompson, and P. Ruddy, "Compressing IP/UDP/RTP Headers on links with high delay, packet loss and reordering", *IETF-draft*, March 2002.

[104] Z. Liu, and K. Le, "0-byte Support for R-mode in Extended Link-layer Assisted ROHC Profile", *IETF-draft*, May 2002.

[105] S. Dawkins, G. Montenegro, M. Kojo, and V. Magret, "End-to-end Performance Implications of Slow Links", *IETF-draft*, July 2001.

[106] G. Pelletier, "Robust Header Compression (ROHC): Profiles for UDP Lite", *IETF-draft*, August 2002.

[107] L-A larzon, M. Degermark, and S. Pink, "The UDP Lite Protocol", *IETF-draft*, August 2001.

[108] H. Hannu, J. Christoffersson and K. Svanbro, "Application signaling over cellular links", IETF-draft, January 2002.

[109] Y. Hi, and V. O-K. Li, "Satellite-Based Internet: A Tutorial", *IEEE Communications Magazine*, March 2001.

[110] C. Perkins, and J. Crowcroft, "Effects of Interleaving on RTP Header Compression", Proceedings of IEEE Infocom 2000, Tel Aviv, March 2000.

[111] D.R. Wisley, "SIP and conversational internet applications", *BT Journals, Vol.19, No.2*, April 2001.

[112] L-A. Larzon, M. Degenmark, and S. Pink, "UDP Lite for Real Time Multimedia Applications", *Proceedings of the Sixth IEEE International Workshop on Mobile Multimedia Communications*, 1999.

[113] "RTP/UDP/IP Header Removal and Construction for Voice over GERAN PS Domain Speech Calls", *3GPP TSG GERAN # 2, GP-000512*, November 2000.

[114] Y. Saifullah, "Common Radio Access Protocols Issues and Requirements", *IETF-draft*, 2000.

[115] H. Hannu, "Signaling Compression Requirements & Assumptions", *IETF-draft*, November 2001.

[116] H. Hannu, J. Christoffersson, C. Clanton, K. Svanbro, "Signaling Compression", IETF-draft, November 2001.

[117] L-E. Jonsson, and G. Pelletier, "Robust Header Compression (ROHC): A link-layer assisted profile for IP/UDP/RTP", IETF-draft, April 2002.

[118] "A. Bria, F. Gessler, O. Queseth, R. Stridh, M. Unbehaun, J. Zander, and M. Flament, "4th –Generation Wireless Infrastructures: Scenarios and Research Challenges", IEEE Personal Communications, December 2001.

[119] "R. Prased, and T. Ojanpera, "An Overview of CDMA Evaluation toward Wideband CDMA", IEEE Communicatios Surveys, Vol.1, No.1, 1998.

[120] M. A. West, L. W. Conroy, R. E. Hancock, R. Prince, and A. H. Surtees, "IP Header and Signalling Compression for 3G Systems", IEEE VTC'02, 2002.

[121] Resenberg, Schulzrinne, Camarillo, Johnston, Peterson, Sparks, Handley, and Schoeler, "SIP: Session Initiation Protocol", IETF-draft, October 2001.

[122] A. j. Fidler, G. Hernandez, M. Lalovic, t. Pell and I. G. Rose, "Satellite- a new opportunity for broadband applications", *BT Technology Journal Vol.20, No.1,* January 2002.

[123] "Packet Data Convergence Protocol (PDCP) Specification", Release 5, 3GPP TS 25.323 v5.1.0, June 2002.

[124] A. B. Downey, "Using Pathchar to estimate Internet link characteristics", *ACM SIGCOMM'99*, 1999.

[125] "V. Brazauskas, and R. Serfling, "Robust and Efficient Estimation of the Tail Index of a Single-Parameter Pareto Distribution", North America Actuarial Journal, February 2000.

[126] Q. Zang, Y-Q. Zhang, and W. Zhu, "Resource Allocation for Audio and Video Streaming over the Internet", IEEE Internetional Symposium on Circuits and Systems, Geneva, May 2000.

[127] Technical Specification Group Core Network, "Mobile radio interface signaling layer 3" 3GPP TS 24.007 v3.6.0, December 2000.

[128] Technical Specification Group (TSG) RAN, "Delay Budget within the Access Stratum", Release 4, 3GPP TR 25.853 v4.0.0, March 2001.

[129] Technical Specification Group (TSG) RAN, "MAC Protocol Specification", Release 4, 3GPP TS 25.321 V4.3.0, December 2001.

[130] Technical Specification Group (TSG) RAN, "Radio Link Control (RLC) Protocol Specification", 3GPP TS 25.322 v3.10.0, March 2002.

[131] Technical Specification Group (TSG) Services and System Aspects and Service Aspects, "Services and Service Capabilities", *3G TS 22.105 v4.1.0,* January 2001.

[132] M. Reid, "Multimedia conferencing over ISDN and IP networks using ITU-T H-series recommendations: architecture, control, and coordination", *Computer Networks Magazine*, 1999.

[133] K. Fujimoto, S. Ata, and M. Murata, "Statistical Analysis of Packet delays in the Internet and its application to playout Control for Streaming Applications", *IEISE Transaction on Communications, Vol.E00-B, No.6,* June 2001.

[134] D. S. Eom, M. Sugano, M. Murata, and H. Miyahara, "Improving TCP Performance by Packet Buffering in Mobile IP Based Networks", *12th annual Conference, JSAC99,* 1999.

[135] H. Kaaranen, A. Ahtiainen, L. Laitinen, S. Naghian, and V. Niemi, "UMTS Networks: Architecture, Mobility and Services", *WILEY,* 2001.

[136] H. Holma, and A. Toskala, "WCDMA for UMTS; Radio Access for Fhird Generation Mobile Communications", *WILEY,* 2000.

[137] F. Halsall, "Data Communications, Computer Networks and Open Systems", *Addison-Wesley,* 1996.

[138] M. Tanner , "Practical Queueing Analysis", *McGRAW-HILL,* 1995.

[139] Kleinrock, "Queueing systems volume I and II Theory", John wiley & sons, 1975.

[140] A. S. Tanenbaum, "Computer Networks", *Prentice-Hall Inc.*, 1996.

[141] P. Larsson and H. Olofssan, "Performance evaluation of different frequency reuse patterns and channel schemes in GPRS", *Proceedings of the IEEE 48th Vehicular Technology Conference, VTC'98, Vol.1, pp.139-143,* Ottaws, Canada, 18-21 May 1998

[142] H. Granbohm and J. Wiklund, "GPRS-General Packet Radio Service", Ericsson Review, NO.2, pp.82-88, 1999.

[143] ETSI/SMG2, "EGPRS 8PSK Receiver Performance", *Tdoc SMG2, EDGE 1444/99,* May 1999.

[144] 3GPP, Technical Specification, 3rd Generation Partnership Projects, Group (TSG) Services and System Aspects and Service Aspects, "QoS Xconcept and Architecture (Release 4)", *3GPP TS 23.107 v4.0.0,* December 2000.

[145] 3rd-Generation Partnership Project (3GPP) Release 5 requirements on the Session Initiation Protocol (SIP) "draft-ietf-sipping-3gpp-r5-requirements-00.txt", *Internet Draft,* October 2002.

[146] Postel, J., Reynolds, J., "File Transfer Protocol (FTP)", *STD 9, RFC 959*, October 1985

[147] R. Fielding, J. Gettys, J. Mogul, H Frystyk, L.Masinter, P. Leach, T. Berners-Lee , "Hypertext Transfer Protocol -HTTP/1.1", *RFC 2616* June 1999.

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

[149] M. Handley, C. Perkins and E. Whelan, ``Session Announcement Protocol", *RFC 2974*, October 2000.

[150] M. Handley, V. Jacobson, C. Perkins, "SDP: Session Description Protocol", *Internet Draft,* November 2002.

[151] L.Westberg, G. Karaguannis, and D. Partain, "RSVP: Resource reservation Protocol", *Internet Draft*, October 2002.

[152] Draft ITU-T Recommendation H.263, "H.263: Video coding for low bitrate communication", Jan 1998.

[153] CCITT Recommendation H.261, "H.261: Video Codec for audiovisual services of p864 kbits/s", *COM XV-R 37-Er*, 1990.

[154] R. Koenen, "Overview of the MPEG-4 Standard", In Doc N3747

[155] S.Fukunaga, Y.Nakaya, S.H.Son, and T.Nagumo, ..., "MPEG-4 Video Verification Model Version 16.0", Number N3312, ISO/IEC JTCI/SC29?WG11, Nordwijkerhout, March 2000.

[156] ITU-T Recommendation H.323v3, "Packet-based multimedia communications systems", ITU-T 1999

[157] ITU-T Recommendation Q.931,

[158] ITU-T Recommendation H.225v4, "Call signalling protocols and media stream Packetisation for packet based multimedia communication systems", ITU-T 1999.

[159] ITU-T Recommendation G.712, "Transmission performance characteristics of pulse code modulation channels", ITU-T November 1996.

[160] Leonardo Chiariglione, "Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s", ISO/IEC JTC1/SC29/WG11, MPEG96, June 1996.

[161] Leonardo Chiariglione, "Generic coding of moving pictures and associated audio information", ISO/IEC JTC1/SC29/WG11, MPEG 00, October 2000.