PERFORMANCE EVALUATION AND OPTIMIZATION OF MPEG-4 VIDEO STREAMING OVER CDMA WIRELESS NETWORKS

by

YING LUO

B.Eng., Southeast University, P.R.China, 1989

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

March 2004

© Ying Luo, 2004

Library Authorization

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Ying Luo

Name of Author (please print)

21/04/2004

Date (dd/mm/yyyy)

Title of Thesis:	Performance Evaluation and Optimaization of MPEG-4 Video
Streaming over CDMA Wireless Networks	

Degree: Master of Applied Science

Year: 2004

Department of Electrical and Computer Engineering The University of British Columbia

Vancouver, BC Canada

Abstract

With the advent of next-generation multimedia technologies such as the low bit rate MPEG-4 codec, multimedia streaming over third generation (3G) wireless networks becomes a reality. To deal with the error prone and bandwidth limited nature of wireless networks, this thesis presents a novel error control scheme called Adaptive Delay-constrained Selective Repeat ARQ (ADSR-ARQ) in the radio link layer. ADSR-ARQ is capable of dynamically switching between ARQ and non-ARQ mode and adjusts the retransmission attempts based on the predefined transmission buffer length thresholds to satisfy the overall quality of service (QoS) requirements of video streams. An analytical model is created to analyze the feasibility and efficiency of this measurement-based ARQ scheme.

To evaluate the performance of the proposed ADSR-ARQ error control scheme under wireless multimedia environment, as well as to investigate the behavior of video streaming applications under various wireless network conditions, we build a LAN-based real-time wireless video streaming testbed. This testbed includes a wireless emulator, an MPEG-4 streaming server and a client. The wireless emulator implementation is based on the NS2 simulation platform operating in emulation mode to simulate the behavior of CDMA wireless networks, such as packet loss and delay, etc. A two-state Markov chain channel model is implemented in the wireless emulator to model the burst error behavior in wireless networks.

In this thesis, end-to-end delay/jitter and throughput of MPEG-4 video transmission are measured under different error conditions over the testbed. The tradeoff between packet loss and delay is evaluated. From the theoretical analysis and experiments results, with an optimal design, our proposed innovative ADSR-ARQ error control scheme can effectively decrease packet losses and improve the channel utilization significantly under specified delay constraints. The ADSR-ARQ scheme outperforms the conventional SR-ARQ scheme with throughput improvement and delay decrease for transmitting MPEG-4 video stream over wireless networks. We also employ a pseudo Robust Header Compression (ROHC) function to demonstrate how the header compression scheme improves the wireless bandwidth efficiency. Furthermore, for future streaming media applications, this wireless networks.

Table of Contents

:

Abstractii
Table of Contentsiii
List of Tablesvii
List of Figuresviii
Glossaryx
Acknowledgmentsxiii
Chapter 1: Introduction1
1.1 Challenges and Related Work
1.1.1 Properties of Streaming Video
1.1.2 Characteristics of Wireless Environment
1.1.3 Issues of Video Streaming over Wireless Network
1.1.4 Related Work
1.1.4.1 Error Recovery Schemes
1.1.4.2 Packet Scheduling
1.1.4.3 Queue Management
1.2 Motivation and Objectives
1.3 Summary of Contributions
1.4 Overview of the Thesis
Chapter 2: Video Streaming and MPEG-413
2.1 Video Streaming
2.1.1 Streaming Architecture
2.1.2 Streaming Protocols
2.1.2 Streaming Proceeds
2.1.2.1 ICI
2.1.2.3 RTP
2.1.2.4 RTSP
2.2 MPEG-4 Standard16

2.2	2.1	Overview	16
2.2	2.2	MPEG-4 Architecture	17
2.3	DM	IIF Communication Model	17
2.4	MP	EG-4 Streaming Server	19
2.4	1.1	System Architecture	19
2.4	1.2	Streaming Server	19
2.5	Vic	leo Format and Color Theory	20
2.5	5.1	Video Format	20
2.5	5.2	Video Color Space	20
2.5	5.3	RGB to YUV Conversion	21
Chapte	er 3:	Wireless Network Emulator Implementation	23
3.1	Wi	reless Communication Overview	23
3.2	IS-	95 & CDMA2000 Standard	24
3.2	2.1	IS-95 CDMA Standard	24
3.2	2.2	CDMA2000 Standard	24
3.3	Wi	reless Channel Error Models	27
3.4	Wi	reless Channel Simulation Tools and Principles	30
3.5	CD	MA Wireless Network Emulator Design and Deployment	31
3.5	5.1	Emulation Environment	31
3.5	5.2	Wireless Emulator System Design	32
3.5	5.3	Wireless Emulator Network Stack	33
3.5	5.4	Wireless Emulator Functionality	35
	3.5.4.	I Channel Model	35
	3.5.4.2	2 Link Layer Error Recovery Scheme	35
	3.5.4.3	3 Robust Header Compression Scheme	36
	3.5.4.4	4 Traffic Shaping	36
	3.5.4.5	5 Priority Scheduling Scheme	37
Chapte	er 4:	Proposed Error Control Scheme and ROHC	38
4.1	Co	nventional Transmission Error Control Techniques	38
4.1	1.1	Forward Error Correction	38
4.1	1.2	Automatic Repeat Request	39
4.2	Ad	aptive Delay-constrained Selective Repeat ARQ Scheme	41

4.2.1	ADSR-ARQ Scheme Design	41
4.2.1.1	Three Service Policies	42
4.2.1.2	Components of ADSR-ARQ	43
4.2.1.3	ADSR-ARQ Policy Based on Three-level Thresholds	44
4.2.2	Analytical Model	45
4.2.3	ADSR-ARQ Implementation in The Simulation Model	47
4.3 Esti	mation of Buffer Length Thresholds in ADSR-ARQ	50
4.3.1	End-to-End Delay	50
4.3.2	Three-level Buffer Length Thresholds	50
4.3.2.1	Scenario 1: (Th2 <l<=th3)< td=""><td> 52</td></l<=th3)<>	52
4.3.2.2	Scenario 2: (Th1 <l<=th2)< td=""><td> 53</td></l<=th2)<>	53
4.3.2.3	Scenario 3: (L<=Th1)	54
4.4 Rob	ust Header Compression (ROHC) Scheme	54
Chapter 5: 1	Performance Evaluation	57
5.1 Met	hodology	57
5.1.1	Experiment Goals	
5.1.2	Experiment Setup	58
5.1.2.1	MPEG-4 Streaming Server	59
5.1.2.2	Wireless Emulator	59
5.1.2.3	Client Terminal	59
5.1.3	Measurement Methods	60
5.1.4	Experiment Assumptions	61
5.2 Exp	eriment Results	62
5.2.1	Channel Error Model verification	63
5.2.1.1	Packet Error Rate	63
5.2.1.2	Error Pattern	64
5.2.2	ROHC Performance Evaluation	66
5.2.3	Influence of Network Bandwidth	69
5.2.4	ADSR-ARQ Error Control Scheme Evaluation	72
5.2.4.1	Delay Jitter	73
5.2.4.2	End-to-End Delay and Throughput	
5.2.4.3	Packet Loss vs. Video Quality	
5.2.5	Summary of Experiments	84

Chapter 6:	Conclusions and Future Work	86
Bibliograph	V	88

,

List of Tables

.

Table 2.1. Network Protocol Stack for Delivering Multimedia	14
Table 4.1. RLC Layer PDU Header Fields	48
Table 5.1. Packet Error Rate vs. Bit Error Rate	64
Table 5.2. Experiment Parameters	72
Table 5.3. Packet Error Rate vs. Video Frame Loss	82

List of Figures

Figure 5.16.	Snapshots of Frame 65 in the "Suzie" Sequence	82
Figure 5.17.	PSNR Comparison for the "Suzie" Sequence	84

Glossary

Acronyms:

.

2G	Second Generation	
3G	Third Generation	
3GPP	Third Generation Partnership Project	
ACK	Acknowledgement	
ADSR-ARQ	Adaptive Delay-constrained Selective Repeat ARQ	
AQM	Active Queue Management	
ARQ	Automatic Repeat Request	
BER	Bit Error Rate	
BPF	Berkeley Packet Filter	
CBR	Constant Bit Rate	
CDMA	Code Division Multiple Access	
CIF	Common Intermediate Format	
DDSP	DMIF Default Signaling Protocol	
DMIF Delivery Multimedia Information Framework		
DS-CDMA	S-CDMA Direct Sequence CDMA	
ESP	Encapsulating Security Payload	
FDMA	Frequency Division multiple Access	
FEC	Forward Error Correction	
GOP	Group of Pictures	
GPRS	General Packet Radio Service	
GSM	Global System for Mobile Communications	
IMT-2000	International Mobile Telecommunications-2000	

1895	Interim Standard 95	
IETF	Internet Engineering Task Force	
LAC	Link Access Control	
LAN	Local Area Network	
LL	Link Layer	
MAC	Medium Access Control	
MPEG	Moving Pictures Expert Group	
MSE	Mean Squared Error	
NACK	Negative Acknowledgement	
NIC	Network Interface Card	
NS2	Network Simulator 2	
OSI	Open System Interconnection	
PER	Packet Error Rate	
PDU .	Protocol Data Unit	
PSNR	Peak Signal-to-Noise Ratio	
QCIF	Quarter Common Intermediate Format	
QoS	Quality of Service	
RFC	Request For Comments	
RGB	A color space, represents Red, Green, Blue	
RLC	Radio Link Control	
RLP	Radio Link Protocol	
ROHC	Robust Header Compression	
RTCP	Real Time Transport Control Protocol	
RTP	Real Time Transport Protocol	

RTSP	Real Time Streaming Protocol	
RTT	Round Trip Time	
ТСР	Transmission Control Protocol	
TDMA	Time Division Multiple Access	
TIA	Telecommunications Industry Association	
UDP	User Datagram Protocol	
UMTS	Universal Mobile Telecommunications System	
URL	Uniform Resource Locator	
VBR	Variable Bit Rate	
YUV	A color space, Y stands for the luminance, U for chrominance and V for saturation	

.

Acknowledgments

I would like to express my sincere gratitude to my thesis supervisor, Dr. Victor Leung for his invaluable guidance, enthusiastic encouragement and support in every stage of my thesis research. He provided me many opportunities for growth: from reading papers, writing a survey, proposing creative ideas, to getting through the inevitable research setbacks, and finishing the thesis. He opened the door for me to enjoy the research work.

I am also grateful to my thesis co-supervisor, Dr. Hussein Alnuweiri for his advice and helpful discussions on multimedia coding technologies. In particular, he provided me with his previous students' source codes of MPEG-4 streaming server, which was integrated into our wireless multimedia testbed. Many thanks to Dr. Panos Nasiopoulos and his student, Kemal Ugur, for their useful information about video encoding/decoding and relevant discussion.

I would like to thank my lab- and team-mate Zhanping Yin for his explanation of the previous work of wireless emulator and suggestions to the new work. To the fellow group members of the Communications Lab, I would like to extend my gratitude for their friendship, encouragement, as well as positive discussions. I thank all my friends who have ever helped and supported me during my studies. Many thanks to Paul Chu and Terry Lum for their proofreading of my writing. Thanks to TELUS Mobility, BC ASI and NSERC for funding this project.

Finally, I want to express my great appreciation to my beloved family, especially my parents and my brother, for their love, encouragement and support. Without their support, my graduate study would have been impossible. I dedicate this thesis to my parents.

With the rapid growth of the Internet, new technologies have brought sound, video, animation and still graphics into the World Wide Web. Over the last few years there has been a dramatic improvement in the quality of IP-based network multimedia applications. Both real time and on-demand multimedia can now be created, served and played over the Internet. At the same time there is an increasing trend toward accessing the Internet and Web from wireless mobile devices.

Real-time transport of live video or stored video is the predominant part of multimedia communications in today's Internet. There are two main types of digital video through the Internet: downloadable content vs. streaming content. Conventionally, digital video can only be played after downloading the entire file to one's computer. Digital video files are massive to store and take a long time to download. In order to solve this problem, streaming technologies have been developed to receive the data from a server and play the video at the same time. With the rapid development of high-speed inter-networks, the latter has become more popular [1]. However, streaming technologies for transmitting real-time audio and video media over the Internet pose many challenges in various areas including media compression, application quality of service (QoS) control, continuous media distribution services, streaming servers, media synchronization mechanisms, and protocols for streaming media.

Today's second generation (2G) cellular telephony networks, such as Global System for Mobile Communications (GSM), typically provide 10-15 Kbps data transmission rate, suitable for compressed speech, but too little for motion video. Fortunately, the third generation (3G) wireless systems, e.g. IMT-2000 and UMTS [2][3], are becoming available, which, together

with the advent of low bit-rate video compression technology, will lead to a new era of wireless multimedia services in the near future [4]. By offering data transmission rates up to 384Kbps for wide-area coverage and 2Mbps for local-area coverage, 3G wireless systems are expected to provide higher bit-rate and better quality data services suited for transmitting multimedia information. Code Division Multiple Access (CDMA) has become one of the leading techniques employed in 3G standards for digital cellular and personal communications services because of its advantages in capacity and performance.

MPEG-4 is one of the emerging next-generation, global multimedia standard, which is promising to become the leading compression codec for the delivery of audio and video to portable voice-data devices. The MPEG-4 Simple Profile provides commonly useful tools. It starts with H.263, the highly efficient video-compression standard developed for videophones. Of the many MPEG-4 features, low-bit-rate video is most generally useful, which makes it well suited for bandwidth-limited wireless audio-visual applications.

Building on the rapid growth and commercial success of mobile communications and multimedia communications, wireless multimedia services will likely find widespread acceptance in the next decade. Wireless multimedia streaming is envisioned to become an important service over packet-switched 2.5G and 3G wireless networks, which consists of a streaming server and a wireless client. Delivery of real-time video stream typically has QoS requirements, e.g., bandwidth, delay and error requirements. However, wireless channels are unreliable and the channel bandwidth varies with time, which may cause severe degradation to video quality. Therefore, improving the performance of streaming video over wireless networks has received tremendous attention from both academia and industry in recent years.

1.1 Challenges and Related Work

1.1.1 Properties of Streaming Video

Due to its real-time nature, video streaming typically has stringent QoS requirements, such as bandwidth, delay/jitter and packet loss:

- **Bandwidth:** Video streams are usually modeled by variable-bit-rate (VBR) traffic sources. For a real-time application, an acceptable bandwidth would be the peak rate. If a bandwidth equal to the peak rate is allocated to the video stream, each packet will arrive at the receiver early enough so that it can be played back in its proper time instant. Otherwise, some packets will not arrive in time for playback. In such cases, a playback buffer will be used to store the packets. In general, if the bandwidth is much less than the average media date rate, the quality of video stream transmission is not acceptable.
- Delay/Jitter: One main problem for streaming video is its strict delay-constrained requirement. Since the packetized video data must be received before its decoding deadline, any data that is lost or that arrives late will be useless for the decoder and the player. Jitter is the difference of inter-arrival time between consecutive packets. Delay jitter is the primary enemy of video streaming since it may cause late-received data to be useless and have propagation effects through a certain length of video sequence. The jitter is mainly due to the delay through transmission over networks, such as queuing delay, retransmission delay.
- Packet Loss: Compressed video signals are extremely vulnerable against packet loss. To reduce the bit rate, the video is compressed to remove the redundancy and compression schemes rely on inter-frame coding for high coding efficiency, i.e., they

use the previous encoded and reconstructed video frame to predict the next frame. Therefore, the loss of information in one frame has considerable impact on the quality of the following frames. Lost packets can severely influence the quality of received video streams.

1.1.2 Characteristics of Wireless Environment

Wireless channels have been shown to be bandwidth limited, error prone, and time varying, resulting in significant amount of packet loss and delay. In a wireless environment, the channel conditions change significantly over time in several ways. This is due to factors such as noise, interference, multi-path, and the movement of the mobile host. The typical impediments of wireless environment are as following:

- Bandwidth limited: Compared to wire networks, the capacity of wireless channels is very limited. For 2G and 3G cellular wireless systems, the bandwidth ranges from 10 Kbps to 384 Kbps in wide area network access.
- Error prone: Wireless channels are typically much noisier due to multi-path fading, thus making the bit error rate very high. Transmission errors of a mobile wireless radio channel range from single bit error to burst errors or even an intermittent loss of the connection. In general, a typical cellular radio channel gives, without any improvement, approximately 10⁻³ bit error rate (BER).

1.1.3 Issues of Video Streaming over Wireless Network

Effective video streaming over the Internet, or best-effort packet-switched networks, in general, has remained a challenge due to the mismatch between the quality of service (QoS) required by the transport of compressed video and that offered by the network. In addition,

wireless transmission channels still face a major challenge for streaming media due to their error characteristics [5][6]. Instead of packet losses caused by congestion, in wireless networks random transmission errors prevail. Beyond the limited available bandwidth, wireless multimedia transmission offers a number of technical challenges.

First of all, computer networks were initially designed to carry non-real-time traffic, in particular for remote login, electronic mail, and file transfer. Many design decisions in traditional protocol stacks are based on the assumption in which bounds on delay and jitter are not an issue. As a consequence, traditional networks, such as the current Internet, are not able to handle multimedia traffic well. For example, the Internet is based on TCP/IP protocol stack [7], which contains four layers: the hardware layer, the network layer, the transport layer, and the application layer. Internet Protocol (IP) serves as the network-layer protocol to provide "best-effort" service for all the datagrams it carries without QoS guarantee. Transmission Control Protocol (TCP) is a reliable connection-oriented protocol that provides reliable and ordered delivery based on retransmissions. Its long delay cannot match the QoS requirements of delay sensitive real-time applications. User Datagram Protocol (UDP) is an unreliable, connectionless protocol that is usually used to transport multimedia stream. Another challenge posed by UDP is how to control the end-to-end QoS over the unreliable transport protocol.

Secondly, the wireless networks cannot provide a guaranteed QoS because of the high bit error rates [8]. Packet losses greatly degrade the video stream quality. Error control schemes are necessary in these error prone network environments. The classical technique to combat transmission errors is Forward Error Correction (FEC), but its effectiveness is limited due to widely varying error conditions. Closed-loop error control techniques like Automatic Repeat Request (ARQ) [9] have been shown to be more effective than FEC and successfully applied to

wireless video transmission [10][11]. Retransmission of corrupted data frames, however, introduces additional delay and delay jitter, which might be unacceptable for real-time video streaming applications. As a result, transmission errors cannot be avoided with a mobile radio channel, even when FEC and ARQ are combined. Although error concealment is an effective way to recover the loss information errors at the decoder, from network service providers' point of view, efficient error control schemes are still necessary for video streaming transmission over wireless networks.

Furthermore, video compression techniques are typically not designed with the transport channel characteristics in mind. Thus, such network-ignorant compression techniques cannot accommodate the dynamic nature of best-effort packet networks in terms of bursty losses and time-varying bandwidth and latency.

1.1.4 Related Work

Transporting video streams over the wireless networks typically has bandwidth, delay and loss requirements, which cannot be adequately supported by the current Internet. To address those problems and challenges, there have been other works dealing with the end-toend performance improvement of video streaming over wireless networks. Generally, there are three categories of approaches, such as content-based (also called source-based) approaches focusing on source coding techniques and media-aware solutions, network-based (also called channel-based) approaches dealing with channel coding algorithms and relevant network protocols, and source-channel joint approaches.

From content providers point of view, several frameworks and solutions for efficiently transporting video streams have been proposed in research papers [12] [13], which focused on

source compression algorithms, media scalability, rate control, packetization, and error resilience etc. With the emergence of new multimedia standards, MPEG-4 is apparently becoming the mainstream technique for mobile video use. In our research, we are more interested in network-based solutions in the wireless domain. From those published work, most researchers focused on the following three areas:

1.1.4.1 Error Recovery Schemes

Some modified ARQ schemes were proposed in [14][15][16]. In [14], the authors proposed a QoS aware SR-ARQ, which dealt with the layer property of MPEG frames and considered different QoS requirements of different data sections within a video stream. For example, in a layered encoded video application, if lower layers have minor errors, they will not adversely affect the overall perceived video quality. Furthermore, the packets related to I, P, B frames are treated differently according to their significance. A Buffer-controlled Retransmission-based Error Control (BREC) scheme was presented in [15]. The retransmission was based on the playback buffer position to prevent it from the underflow and overflow. In [16], a conditional retransmission based on error concealment strategy was proposed. That is, some packets may not be worth re-transmitting if the error concealment at the decoder can do a good job. All these retransmission-based schemes were media-aware solutions, that had to have certain knowledge about the media contents, such as having knowledge of encoded streams, recognizing the packet types, dealing with media decoder or playback buffer etc. In addition, all the above schemes were verified by computer simulations.

1.1.4.2 Packet Scheduling

For the performance improvement of video streaming over wireless channels, some packet scheduling algorithms were proposed by other researchers [17]. The main idea of packet

scheduling algorithms for video streaming over wireless channels is to apply different deadline thresholds to video packets with different importance. The importance of a video packet is determined by its relative position within its group of pictures (GOP) and motion-texture context. For example, the loss impact of earlier frames within a GOP on video quality is much greater than that of the later frames within it. Furthermore, the motion part is more important than the texture part within a frame. Thus it is desirable to send the more important parts of video prior to the less important ones in order to ensure more opportunities for retransmissions in case of channel errors. This algorithm also requires the networks to be aware of the media content type. This algorithm was evaluated by computer simulations.

1.1.4.3 Queue Management

Research [18] presented an Active Queue Management (AQM) mechanism to limit the transmission queue length by eliminating expired packets at sender. The purpose of this mechanism is to prevent the expiring packets from occupying the limited wireless bandwidth. At the same time AQM minimizes the queuing delay which usually affects the performance of wireless video communication significantly. The AQM scheme was verified by computer simulations as well.

1.2 Motivation and Objectives

The wireless network is a much less stable communication environment than the wired network. Traffic intensity and bandwidth vary much more in the wireless network. Low bandwidth and high error rates are common characteristics in wireless communication, and disconnections are frequent. Therefore, wireless networks pose a significant challenge for transmitting video streams.

Although several published research papers in the wireless multimedia area claimed substantial performance gains by using their proposed solutions, most of those solutions were content-aware and were verified by computer simulations. For example, in the simulation, video traffics are usually simulated by video trace files. In addition, those theoretical assumptions based on the simulations might not satisfy the real world conditions all the time. From the wireless service providers' point of view, such approaches are difficult to be deployed in practice since most network service providers do not deal with the media contents directly. Therefore, considering the practical significance of evaluating and improving the performance of MPEG-4 video streaming over wireless channel, in this thesis, we propose to use a live MPEG-4 streaming server with emulated CDMA wireless channel to build a LAN-based real-time wireless video streaming testbed.

Network emulation aims to combine the best properties of both simulation and "live" testing, which can be used to explore efficient end-to-end transport of MPEG-4 video streaming and verify the new proposed error control schemes or other algorithms for performance optimization under different channel models. The wireless video streaming testbed provides an environment which models all radio link layer facets of a real CDMA wireless system to study and investigate the issues and solutions of video stream over wireless networks. To the best of my knowledge, there has not been any significant study or literature on performance evaluation of live MPEG-4 video streaming over emulated wireless environment. Thus, the objectives of this thesis include:

• Performance enhancement

Due to error-prone and bandwidth-limited nature of wireless environments, packet losses significantly degrade the video quality. How to improve the performance of transporting

video streams over wireless networks is one of our main objectives. To overcome the drawback of long delay of the conventional SR-ARQ scheme caused by persistent retransmissions, we propose and implement a new error control scheme in the radio link layer.

• Evaluation of end-to-end performance

To study and investigate how the effects of delay, packet loss and delay jitter on video quality are, as well as evaluate how the wireless Link Layer parameters and protocols affect the performance of MPEG-4 video streaming delivery over wireless networks, a controlled test environment is necessary. Therefore, a wireless video streaming testbed is critical to the endto-end performance evaluation.

• A comprehensive method of algorithm evaluation

To develop and verify appropriate algorithms for wireless streaming media, extensive tests under various loss patterns and network parameters are required. Since these networks are either in field trial or still under development, tests are extremely complex and costly. Off-line simulation is not sufficient, which does not provide accurate insights into the algorithm behavior in a real system. Our wireless video streaming testbed hopes to fill in this gap by providing an environment which models all facets of a real wireless system and provides a complete picture of system behavior with various algorithms. Our proposed error control scheme and the header compression scheme are evaluated over this testbed.

1.3 Summary of Contributions

In general, our research work provides a fundamental contribution to the performance investigation and evaluation of MPEG-4 video streaming over CDMA wireless networks. It covers several aspects: We study queue management and link-layer protocols in the radio

network; We develop channel models to simulate the packet error/loss behavior of wireless networks to investigate how the error prone characteristics of wireless channels affect the video streaming transmission; We also demonstrate the tradeoff between the packet loss and delay; We evaluate the performance of robust header compression scheme to show how this scheme significantly improves bandwidth efficiency in the wireless environment. One of the important contributions of this thesis is that we propose an Adaptive Delay-constrained Selective Repeat ARQ error control scheme to overcome the drawbacks of conventional SR-ARQ scheme for video streaming applications, and improve overall quality of video streaming transmission. Another important contribution is to develop and implement a LAN-based real-time wireless video streaming testbed, which is critical to investigate and identify performance problems.

These contributions are also categorized as following:

• To wireless service providers:

We developed a generic CDMA 3G wireless emulator, which is configurable to provide different wireless network parameters, such as error control schemes, error models, link layer fragment size, bandwidth, packet error rates etc. This emulator can be used to tune and investigate new network protocols or schemes over IP based networks. In addition, our proposed novel ADSR-ARQ error control scheme is network-based, which can be easily deployed in practice by the network service providers to optimize the network parameters and provide a satisfactory service for streaming video delivery.

• To multimedia content creator:

The wireless video streaming testbed aims at providing an environment which models all facets of a real wireless multimedia system and provides a complete picture of system behaviour with various algorithms. Since our CDMA wireless emulator is content independent, it can be applied to any kind of multimedia applications to test the new multimedia compression algorithms and other schemes.

1.4 Overview of the Thesis

١,

The remainder of this thesis is organized as follows. Chapter 2 provides a general description of video streaming and MPEG-4 related technologies. Next, an MPEG-4 streaming system is introduced as well. In Chapter 3, we first give an overview of wireless communication and introduce IS95 and CDMA2000 cellular standards. Then, the channel models and wireless emulator design and implementation are discussed in detail. Afterwards, after a short review of traditional error control schemes, the proposed novel ADSR-ARQ error control scheme and its analytical model, as well as the Robust Header Compression scheme is described in Chapter 4. In Chapter 5, first, we introduce the methodology including the wireless video streaming testbed set up and experiment assumptions. Then, the performance evaluation of MPEG-4 video streaming over CDMA wireless networks is presented and analyzed in detail, which include error model verification, ROHC performance evaluation and the proposed ADSR-ARQ performance evaluation. Finally, Chapter 6 gives the conclusion from this work.

In this Chapter, a basic introduction of video streaming technology including system architecture and protocols is given in Section 2.1. Then, an overview of MPEG-4 standard and its components is presented in Section 2.2. The MPEG-4 DMIF communication model is introduced in Section 2.3. Afterwards, an MPEG-4 streaming server is described in Section 2.4. Finally, Basic video format and color theory is stated in Section 2.5.

2.1 Video Streaming

2.1.1 Streaming Architecture

Figure 2.1 depicts a typical wireless video streaming system, which consists of a wireless domain and a wired domain. The basic components in this system include a video streaming server, a mobile client, a transcoding/streaming proxy and a base station. In our research, we are more interested in the wireless domain. Thus, the Internet "cloud" and the transcoding/streaming proxy are not discussed in this thesis.



Figure 2.1. Wireless Video Streaming System

2.1.2 Streaming Protocols

The streaming protocol is a crucial component to deliver multimedia across wireless networks. Current end-to-end video streaming applications mostly rely on TCP and UDP as transport protocols for video packet delivery over the Internet [19]. For the delivery of video streams with synchronized audio from a server to a terminal, IETF recommends the use of methods based on RTP and RTSP [20]. Table 2.1 depicts the network protocol stack for delivering multimedia.

Application Control	Audio, Video,	
Commands	Sender/Receiver	
	Reports	
RTSP	RTP/RTCP	
ТСР	UDP	
IP		
Radio Link/Data Link		
Physical Layer		

Table 2.1. Network Protocol Stack for Delivering Multimedia

2.1.2.1 TCP

Transmission Control Protocol (TCP) [21] is a reliable connection-oriented protocol which allows a byte stream originating on one machine to be delivered without error to any other machine on the Internet. TCP uses retransmission to recover lost packets. Since TCP retransmission introduces delay, it is not acceptable for streaming applications with stringent delay requirements.

2.1.2.2 UDP

Unlike TCP, User Datagram Protocol (UDP) [22] is a lightweight, unreliable, connectionless transmission protocol which is more suitable for real-time video streaming since

it is not hindered by any congestion control or retransmission mechanisms [23]. However, UDP does not guarantee packet delivery, the receiver needs to rely on the upper layer to detect packet loss.

2.1.2.3 RTP

Real Time Transport Protocol (RTP) is the Internet-standard protocol (RFC 1889, 1890) designed to provide end-to-end transport functions for supporting real-time applications. RTP consists of a data and a control part called RTCP. The data part of RTP is a thin protocol providing support for applications with real-time properties such as continuous media (e.g., audio and video), including timing reconstruction, loss detection, security and content identification. RTCP is a companion protocol with RTP and is designed to provide Quality of Service (QoS) feedback from receivers indicating packet loss, jitter and round trip time. RTCP supports synchronization of different media streams as well.

2.1.2.4 RTSP

Real Time Streaming Protocol (RTSP), based on the document RFC 2326, is a real-time streaming delivery standard. RTSP is a session-oriented protocol for control and delivery of real-time media, which is transported over TCP between streaming media client and server. RTSP provides a text-based set of commands for transmission control. The RTSP command set includes actions such as describe, setup, play, stop, record, and teardown etc. These commands are usually communicated from client to server to initiate a particular action; hence, the session is controlled. RTSP establishes and controls either a single or several time-synchronized streams of continuous media such as audio and video. It does not deliver the continuous media stream itself.

2.2 MPEG-4 Standard

2.2.1 Overview

MPEG-4 is the successor of the audio and video compression standards MPEG-1 and MPEG-2, which were developed by the MPEG (Moving Pictures Expert Group, part of the International Standards Organization ISO). MPEG-4 provides a broad framework for creating, representing, distributing, and accessing digital audiovisual content. The MPEG-4 basic coding principles and the associated functionalities show that this new coding standard can be very flexible in adapting to different transmission and decoding conditions, such as different bit rates and different error conditions. This makes MPEG-4 the ideal and timely audiovisual coding technology to implement a whole new range of multimedia applications over wireless channels, such as PCS and IMT-2000/UMTS.

MPEG-4 is a digital multimedia standard with associated protocols for representing, manipulating and transporting natural and synthetic multimedia content (i.e. audio, video, data) over a broad range of communication infrastructures. Of the many MPEG-4 features, low-bit-rate video is most generally useful, which makes it well suitable for mobile video devices over 3G wireless links [24]. Some of MPEG-4's capabilities, such as the ability to create new content on a stored video image by transferring only a few bits to the device, are revolutionizing the way that voice and data will be transmitted on the future networks. MPEG-4 has the following advantages:

- Compression Efficiency
- Content-based interactivity
- Universal access
- Robustness in Error Prone Environments

Error resilience will be supported to assist the access of image and video over a wide

range of storage and transmission media. This includes the useful operation of image and video compression algorithms in error-prone environments at low bit-rates (i.e., less than 64 Kbps). Tools are provided which address both the band limited nature and error resiliency aspects for access over wireless networks.

2.2.2 MPEG-4 Architecture

MPEG-4 uses a client/server model. The generic architecture of an MPEG-4 terminal is depicted in Figure 2.2. It consists of three layers: The Compression Layer, the Synchronization Layer (SL) and the Delivery Layer. The Compression Layer performs media encoding and decoding of Elementary Streams; The Sync Layer is responsible for managing Elementary Streams and their synchronization and hierarchical relations; The Delivery Layer ensures transparent access to content irrespective of delivery technology.



Figure 2.2. MPEG-4 Framework

2.3 DMIF Communication Model

The Delivery Multimedia Integration Framework (DMIF) [25] is a layer whose purpose is to de-couple the delivery from the application and ensure end-to-end signaling and transport interoperability between end-systems. The interface between the application and the delivery layer is called DMIF Application Interface (DAI). The MPEG-4 standard does not enforce any

delivery specifications for media transport across the Internet. There are a number of possible delivery platforms that could be tied into this technology for transport across the Internet. The architecture used for realizing the DMIF corresponds to the recommendations made in part 6 of the MPEG-4 standard and is depicted in Figure 2.3 [25].



Figure 2.3. DMIF Communication Model

DMIF supports three major technologies: interactive network technology (e.g. Internet, ATM, etc.), broadcast technology (e.g. cable, satellite, etc.), and storage technology (e.g. CD, DVD, etc). The delivery technology that this thesis focuses on is the interactive network technology. During remote media access, two communication planes are required: a Data Plane for the transport of media data, and a Control Plane used for media session management. In the case of interactive networks DMIF specifies a logical interface called DMIF Network Interface (DNI). The DMIF standard defines a straightforward mapping of DNI primitives into signaling messages; this mapping is named DMIF Default Signaling Protocol (DDSP). DDSP is in fact a session level protocol for the management of multimedia streaming over generic delivery technologies. It has some similarities to other session layer and streaming control protocols such as RTSP and SIP. DDSP comprises primitives to set-up and tear down sessions as well as individual data channels. The protocol uses binary format for its messages.

2.4 MPEG-4 Streaming Server

2.4.1 System Architecture

The MPEG-4 Steaming System Architecture [26] recommended by the MPEG-4/DMIF standard is depicted in Figure 2.4. This MPEG-4 system is based on the client/server architecture. The Control Plane uses TCP as the transport service for DMIF Default Signaling Protocol (DDSP) messages. The Data Plane uses UDP as the transport service for the delivery of MPEG-4 content. Flexibility in exploiting different transport technologies is one of the advantages this architecture offers.



Figure 2.4. MPEG-4 Streaming System Architecture

2.4.2 Streaming Server

Streaming servers play a key role in providing streaming services. To offer quality streaming services, streaming servers are required to process multimedia data under timing constraints and support interactive control operations such as pause/resume, fast forward and fast backward. Furthermore, streaming servers need to retrieve media components in a synchronous fashion. An MPEG-4 streaming server called Apadana Server [26][27][28] has

been successfully implemented in the LAN (Lab for Advanced Networking) at UBC. This MPEG-4 Streaming Server implementation consists of two layers, an application layer and a DMIF layer. Without using RTP/RTCP, it directly transports the SL-packetized streams on UDP. This MPEG-4 streaming server supports interactive media transmission, which can be used over an IP-based network. The detailed implementation of the Apadana Server exceeds the scope of this thesis. We will integrate this MPEG-4 streaming server into our wireless video streaming testbed for performance evaluation.

2.5 Video Format and Color Theory

2.5.1 Video Format

A video stream is a sequence of video frames. Each frame is a still image. A video player displays one frame after another, usually at a rate close to 30 frames per second (fps). Frames are digitized in a standard RGB format, 24 bits per pixel (8 bits each for Red, Green, and Blue). Quarter Common Intermediate Format (QCIF) image size is used in our experiments. QCIF, a videoconferencing format, specifies data rates 15-30 fps, with each frame containing 144 lines and 176 pixels per line. This is one fourth of the resolution of full CIF. QCIF is optimized for lower bit-rate wireless multimedia systems.

2.5.2 Video Color Space

To analyze video quality, we should understand the video color space. RGB and YUV are the most common color space dealing with computer graphics and video applications. The RGB color space is an additive color space in which red, green, and blue light can be combined in various proportions to obtain any color in the visible spectrum. One common application of the RGB color space is the display of computer graphics on a monitor. Each pixel on the screen

can be represented in the computer's memory as independent values for red, green and blue. The computer display system must have 24 bits to describe the color in each pixel, 256 levels for each color. RGB file format is used in files that are created for use on-screen. In our experiments, we modified the IM1 player source codes to capture the video images originally displaying on computer screen and save into an RGB format file for video quality analysis.

YUV, also known as *YCbCr*, is a color space in which the Y stands for the luminance component (the brightness) and U and V are chrominance (color) components. It is commonly used in video applications. YUV uses RGB information, but it creates a black and white image (luma) from the full color image and then subtracts the three primary colors resulting in two additional signals to describe color. Combining the three signals back together results in a full color image. The YUV format is subsampled. All luminance information is retained. However, chrominance information is subsampled 2:1 in both the horizontal and vertical directions. Thus, there are 2 bits each per pixel of U and V information. This subsampling does not drastically affect quality because the human eye is more sensitive to luminance than to chrominance information. Subsampling is a lossy step. The 24 bits RGB information is reduced to 12 bits YUV information, which automatically gives 2:1 compression. This kind of YUV format is also called 4:2:0 YCrCb. This format is widespread in the video research and codec development community.

2.5.3 RGB to YUV Conversion

In our research, in order to analyze the quality of the received video images, since the video format is usually represented in YUV format, we have to convert the captured RGB format video images to YUV format. YUV can be derived from the following formulations:

$$Y = 0.3R + 0.6G + 0.1B$$

 $U = B - Y$ (2.1)
 $V = R - Y$

OR

$$Y = 0.3R + 0.6G + 0.1B$$

$$U = -0.3R - 0.6G + 0.9B$$

$$V = 0.7R - 0.6G - 0.1B$$
(2.2)

In this thesis, we developed a program called rgb2yuv420.exe to convert the RGB format file into YUV420 format file. Then we used a free tool called ImagePlayer to display YUV video streams by frame, which is useful for the subjective evaluation of the quality of . raw video data in the YUV format. In addition, we developed a program called computePSNR.exe to calculate the quality differences in PSNR frame by frame between the original video sequence and the video sequences obtained after network transport.
Chapter 3: Wireless Network Emulator Implementation

This Chapter first gives an overview of wireless communication in Section 3.1. Then, two North America standards IS95 and CDMA 2000 in 2G and 3G cellular wireless networks are introduced in Section 3.2. Section 3.3 presents the wireless channel models developed in the wireless emulator. Afterwards, the emulation tools NS2 and its principles are stated in Section 3.4. Finally, the wireless emulator design and implementation are presented in detail in Section 3.5.

3.1 Wireless Communication Overview

The global mobile communications market is booming. Wireless cellular telephony has been growing at a faster rate than wired-line telephone networks [29]. The main factor driving this tremendous growth in wireless coverage is that it does not need the setting up of expensive infrastructure like copper or fiber lines and switching equipments. This growth has also been fueled by the recent improvements in the capacity of wireless links due to the use of multiple access techniques (which allow many users to share the same channel for transmission) in association with advanced signal processing algorithms. The Code Division Multiple Access (CDMA) technique has become one of the leading standards in the digital cellular systems and personal communications services industry because of its advantage in capacity and performance [30]. Unlike other multiple access techniques such as Frequency Division Multiple Access (FDMA) and Time-Division Multiple Access (TDMA), which are limited in frequency band and time duration respectively, CDMA uses all of the available time-frequency space. One form of CDMA called Direct Sequence CDMA (DS-CDMA) uses a set of unique signature sequences or spreading codes to modulate the data bits of different users. With the knowledge of these spreading codes, the receiver can isolate the data corresponding to each user by the process of channel estimation and detection. CDMA is already a standard for third generation (3G) wireless networks. Standards such as IS-95 and CDMA2000 are based on CDMA technology.

3.2 IS-95 & CDMA2000 Standard

3.2.1 IS-95 CDMA Standard

An U.S. digital cellular system based on CDMA which promises increased capacity has been standardized as Interim Standard 95 (IS-95) [31] by the U.S. TIA. IS-95 is specified for reverse link operation in the 824-849 MHz band and for the forward link in 869-894 MHz band. A forward and reverse channel pair is separated by 45 MHz. Many users share a common channel for transmission. Each IS-95 channel occupies 1.25 MHz of spectrum on each one-way link. The maximum user data rate is 9600 bits/sec. User data in IS-95 is spread to a channel chip rate of 1.2288 Mchip/s using a combination of techniques. The frame duration is fixed at 20 ms. Depending on the design and operation conditions, this allows for serving 16 to 64 users in the 1.25 MHz bandwidth.

Also, IS-95 uses a different modulation and spreading technique for the forward and reverse links. For instance, the forward traffic link utilizes a convolutional code with a code rate of ½ and BPSK modulation, while the reverse channel employs a convolutional code with a code rate of 1/3 and offset QPSK.

3.2.2 CDMA2000 Standard

CDMA2000 is the third Generation (3G) solution based on IS-95. Unlike some 3G standards, it is an evolution of an existing wireless standard. CDMA2000 supports 3G services

Chapter 3: Wireless Network Emulator Implementation

as defined by the International Telecommunications Union (ITU) for IMT-2000. 3G networks will deliver wireless services with better performance, greater cost-effectiveness and significantly more content. The goal is to access any service, anywhere, anytime from one terminal - true converged, mobile service [32]. CDMA2000 is both an air interface and a core network solution for delivering the services that customers are demanding today. The CDMA2000 standard is evolving to continually support new services in a standard 1.25 MHz carrier. The first phase of CDMA2000, or CDMA2000 1X will deliver average data rates of 144 kbps. Phase two, labeled CDMA2000 1xEV, will support data rates of up to 2 Mbps on a dedicated data carrier and 384 Kbps in a wide area carrier. At the physical layer, the standard allows transmission in 5, 10, 20, 40 or 80 ms time frames. Various orthogonal (Walsh) codes are used for channel identification to achieve higher data rates.

CDMA2000 has a layered structure that is designed to provide voice, packet data and signaling services. Figure 3.1 shows IS-95 and CDMA2000 layered structure.

- At the basic level, CDMA2000 provides protocols and services that correspond to the bottom two layers of the Open System Interconnection (OSI) reference model, i.e., physical and link layer. The physical layer performs coding, interleaving, modulation, and spreading functions for the physical channels.
- The link layer (layer 2) provides protocol support and control mechanisms for data transport services and maps the data transport needs of the upper layers into the specific capabilities and characteristics of the physical layer. The link layer is subdivided into the Link Access Control (LAC) and Medium Access Control (MAC) sublayers. The LAC sublayer performs the functions essential to set up, maintain, and release a logical link connection.



Figure 3.1. TIA IS-95 and CDMA2000 Layered Structure with Protocols

• The MAC sublayer provides a control function that manages resources supplied by the physical layer and coordinates the usage of those resources desired by various LAC service entities. The MAC sublayer is also responsible for delivering the Quality of Service (QoS) level requested by a LAC service. In CDMA2000 the MAC uses Radio Link Protocol (RLP), which provides a highly efficient streaming service that makes a best effort to deliver data between peer entities. The RLP provides both a transparent and nontransparent mode of operation. In the nontransparent mode RLP uses Automatic Repeat Request (ARQ) protocols to retransmit data segments that were not delivered properly by the physical layer. In this mode RLP introduces some delay. In the transparent mode, RLP does not retransmit missing data segments. However, RLP does

maintain byte synchronization between the sender and receiver and notifies the receiver of the missing parts of the data stream. Transparent RLP does not introduce any transmission delay and is useful in implementing voice services over RLP.

• Upper-layer entities provide support for multiple concurrent active sessions with any combination of service type.

3.3 Wireless Channel Error Models

Techniques for modeling and simulating channel conditions play an essential role in understanding network protocol and application behavior. The wireless channel is error-prone and varies with time, which results in burst errors. In our CDMA wireless emulator, we observe the channel from the link layer point of view. When video packets are transmitted over wireless channels with burst errors, packet error statistics are more important than bit error statistics to analyze the video streaming delivery performance. In the wireless emulator, we implement two packet erasure channel models to characterize the channel error behaviors. One is random packet erasure channel model. The other is burst packet erasure channel model, a typical analytical error model in wireless network.

In the random error model, the wireless channel is characterized by uncorrelated packet errors. Interleaving techniques are widely suggested to employ in the wireless networks to mitigate the effects of bit burst errors [33]. By incorporating with interleaving techniques, this model provides considerable insights on the error behavior of wireless networks. For simplicity, we implement this model by using a uniform random distribution function.

A well-known class of bit error models, named Gilbert-Elliot two-state Markov chain model, was introduced to characterize fading in the wireless communication channel [34].

27

Chapter 3: Wireless Network Emulator Implementation

Based on the Gilbert-Elliot bit level channel model, several work [35] [36] also derived and demonstrated the equivalent packet level channel models. Therefore, a two-state first order Markov chain is commonly used as packet level error model in most research. In our wireless emulator, we use the two-state Markov chain packet erasure channel model to model the burst packet error/loss behavior. Figure 3.2 shows the two-state Markov chain error model diagram. State "G" represents the channel being in a "good" state, characterized by a very low packet error rate P_G . State "B" indicates the channel operating in a "bad" state, such as a fading condition, with a higher packet error rate P_B . Within one state packet errors are assumed to occur independently from each other. The transition rate from state G to B is denoted by p, while the transition rate from state B to G is denoted by q. The transition rates p and q between the states control the lengths of error bursts. The channel is fully described by the mean packet error/loss rate and the error burst length [37].



Figure 3.2. Two-state Markov Chain Error Model

The steady state probabilities of being in states G and B are π_G and π_B respectively, which are represented in equation (3.1) and (3.2):

$$\pi_G = \frac{q}{p+q} \tag{3.1}$$

$$\pi_B = \frac{p}{p+q} \tag{3.2}$$

The average packet loss rate produced by this channel model is represented in equation (3.3):

Chapter 3: Wireless Network Emulator Implementation

$$\boldsymbol{P}_{loss} = \boldsymbol{P}_{G}\boldsymbol{\pi}_{G} + \boldsymbol{P}_{B}\boldsymbol{\pi}_{B} \tag{3.3}$$

This channel is assumed to have memory which results in burst packet losses. This channel model approximates a typical wireless transmission link where channel outages due to fading or frequent disconnections due to handover cause loss of successive packets. In our wireless emulator we use a Pentium II PC running NS2 (Network Simulation) over FreeBSD to simulate the behavior of CDMA wireless networks. In order to avoid the sophisticated computing that might result in the additional delay on the delay-sensitive video streaming, we use a simplified two-state Markov model in the implementation. We assume $P_G=0$ at "Good" state and $P_B=1$ at "Bad" state. That is, in the state "Good" packets are assumed to be received correctly while in the state "Bad" packets are assumed to be lost. The mean length of these burst losses is determined by the state transition probabilities [38], which can be represented by the following equation (3.4):

$$E[burstlength] = \frac{1}{q}$$
(3.4)

From equation (3.3), the average packet error/loss rate of the simplified two-state Markov error model equals the stationary probability of bad state, show in equation (3.5):

$$P_{loss} = \pi_B = \frac{p}{p+q} \tag{3.5}$$

From equation (3.4) and (3.5), we can easily model the burst error behavior in terms of the mean packet error rate and average error burst length. For example, when a packet is ready to be transmitted over the channel, according to this error model with the given mean packet error rate and expected average burst length, a decision is made if the packet is marked as corrupted at the sender, which will be dropped at the receiver.

3.4 Wireless Channel Simulation Tools and Principles

Simulation is an ideal approach for preliminary evaluation of protocol design as it allows a rapid exploration of a wide parameter space. Emulation refers to the ability to introduce the simulator into a live network. The emulator enables the system to practically operate in real-time. In our CDMA wireless emulator, we use Network Simulator 2 (NS2) running in emulation mode to simulate the CDMA wireless channel behaviors.

NS2 is an open-source simulation tool, which was developed by the Information Sciences Institute at the University of Southern California [39]. It is a discrete event simulator targeted at networking research and provides substantial support for simulation of routing, multicast protocols and IP protocols, such as UDP, TCP, RTP over wired and wireless (local and satellite) networks. It has many advantages that make it a useful tool, such as support for multiple protocols. Additionally, NS2 supports several algorithms in routing and queuing. LAN routing and broadcasts are part of routing algorithms. Queuing algorithms include fair queuing, deficit round-robin and FIFO/Drop –tail, RED etc.

Up to today, NS2 emulation mode just supports "opaque mode", in which the simulator treats network data as un-interpreted packets. In particular, real-world protocol fields are not directly manipulated by the simulator. In opaque mode, live data packets may be dropped, delayed, re-ordered, or duplicated. When using the emulation mode, a special version of the system scheduler, the real-time scheduler, is used. The real-time scheduler implements a soft real-time scheduler that ties event execution within the simulator to real time. NS2 is available on several platforms such as FreeBSD, Linux, SunOS and Solaris, but its emulation mode is only supported under FreeBSD so far.

30

The interface between the simulator and live network is provided by a collection of objects including tap agents and network objects. Tap agents embed live network data into simulated packets and vice versa. The tap agents handle the setting of the common header packet size field and the type field. It uses the packet type PT_LIVE for packets injected into the simulator. Each tap agent can have at most one associated network object. Network objects are installed in tap agents and provide an entry-point for the sending and receipt of live data. Network objects provide access to a live network. There are several forms of network objects, depending on the protocol layer specified for access to the underlying network, in addition to the facilities provided by the host operating system. Generally, network objects provide an entry-point into the live network at a particular protocol layer (e.g. link, raw IP, UDP, etc) and with a particular access mode (read-only, write-only, or read-write). In our wireless emulator, the Pcap/BPF (Berkeley Packet Filter) network objects are used to capture link-layer frames in promiseuous fashion from network interface drivers.

3.5 CDMA Wireless Network Emulator Design and Deployment

3.5.1 Emulation Environment

We use a PC to implement the CDMA wireless network emulator. The configuration of the system are as following:

- Hardware: Pentium II 500MHz, 512MB RAM, 2 NICs PC
- Software: FreeBSD 4.7 Operating System
 Network Simulator 2.1b7a release version
 NOAH extension by Jörg Widmer
 GPRS module/patch by Richa Jain

3.5.2 Wireless Emulator System Design

Our CDMA wireless emulator is simulated by Network Simulator 2 emulation mode with CMU-mobile extensions. The basic wireless network emulation system is depicted in Figure 3.3. The emulator consists of two nodes, one is BS Node for the base station; the other is Mobile Node for the mobile client. Each node has two tap agents attached to a network object respectively. Tap agents inside the emulator are assigned to capture (tap1, tap3) live traffic or inject (tap2, tap4) the processed traffic from the virtual network into the live network. Each tap agent is attached to a virtual network object. The traffic from the entry nodes (bpf1, bpf2) is filtered with respect to the source and destination IP address. Each entry node is connected to an exit node (nd3, nd4) by a simplex link with the bandwidth of CDMA wireless network. Thus, two simplex links are formed between two NICs to simulate the uplink and downlink channels respectively. The captured packets are then fragmented at the link layer and transmitted between the nodes.



Figure 3.3. Wireless Network Emulation System

For simplicity, NS-2 NOAH extension by Jörg Widmer [55] is used to support one

mobile node in our experiments, NOAH is a wireless routing agent that only supports direct communication between the base station and the mobile node without routing. Our wireless channel emulation model and partial source codes are based on [40], which was adapted from the GPRS patch by Richa Jain [52].

3.5.3 Wireless Emulator Network Stack

The network stack for a wireless node consists of a link layer (LL), an ARP module connected to LL, a Radio Link Control (RLC) sublayer, an interface priority queue (IFq), a MAC sublayer, and a network interface (NetIF) which is connected to the channel. The structure of hierarchical wireless node in our emulator is shown in Figure 3.4.



Figure 3.4. Structure of Hierarchical Mobile/BS Node

Chapter 3: Wireless Network Emulator Implementation

• *Link Layer module:* supports data link protocols and mechanisms such as packet fragmentation/reassembly, queue scheduling, link-level retransmissions, piggybacking etc.

• Address Resolution Protocol module: finds and resolves the IP address of the next hop/node into the correct MAC address. The MAC destination address is set into the MAC header of the packet.

• *Radio Link Control Module:* is similar as Link Layer object, which is implemented in our wireless emulator to perform Radio Link Protocol functionalities as well as the proposed new improved ARQ schemes.

• *Interface Priority Queue:* gives a priority to routing protocol packets by running a filter over the packets and removing those with a specified destination address. In our emulator, it gives higher priority to retransmitted packets than those first transmitted ones.

• *Medium Access Protocol module:* provides multiple functionalities such as carrier sense, collision detection, avoidance and QoS control etc.

• *Network Interface:* is an interface for a mobile node to access the channel. Each packet leaving the NetIF is stamped with the meta-data in its header with the information of the transmitting interface such as transmission power, wavelength etc., which is used by the radio propagation model of the receiving NetIF.

• *Radio Propagation Model:* uses Free-space attenuation $(1/r^2)$ at near distance and an approximation to two-ray ground $(1/r^4)$ reflection model at far distance, which decides whether the packet can be received by the mobile node with given distance, transmit power and wavelength.

• Physical Channel - Antenna: Mobile nodes use the unity gain omni-directional antenna.

In our CDMA wireless emulator, we focus on the simulation of Radio Link Layer characteristics to investigate how the wireless radio link behavior affect the MPEG-4 video

Chapter 3: Wireless Network Emulator Implementation

streaming delivery. For example, the live IP packets are captured from real network, which are processed by the CDMA wireless emulator with fragment/assembly, delay, packet loss or retransmission at link layer, then injected back into the real network.

3.5.4 Wireless Emulator Functionality

3.5.4.1 Channel Model

To study and investigate the performance evaluation of wireless multimedia delivery under controlled error conditions, we implement channel error models described in Section 3.3 into the MAC module. Our error model simulates link-level RLP PDU (Protocol Data Unit) errors or losses by marking the PDU's error flag at originating MAC and dropping the erroneous PDU at its destination RLC. For instance, when MAC receives a fragmented PDU from RLC, according to the channel model with the designated packet error/loss rate or burst length, it decides whether the PDU is marked in error or not. At the destination, RLC drops the erroneous PDU as indicated by the error flag. The error model function can be configured ON or OFF and the packet error/loss rate and burst length also can be changed accordingly by the user.

3.5.4.2 Link Layer Error Recovery Scheme

Due to wireless error prone characteristics, error control techniques are necessary to ensure high-quality video stream transmission. Link layer error control schemes are better positioned to handle local issues in a fast and transparent manner, especially for UDP based real time applications, since UDP is a connectionless unreliable transport protocol without error control functions. For performance optimization, we implement our proposed Adaptive Delay-Constraint Selective Repeat ARQ (ADSR-ARQ) scheme in RLC module, which will be described in detail in Section 4.2. The acknowledgement mode also can be configured ON or OFF as a parameter by the user.

3.5.4.3 Robust Header Compression Scheme

In order to improve the bandwidth utilization, IP header compression mechanisms have always been an important topic for the research community to save bandwidth in the IP-based networks. The published research [41] shows the Robust Header Compression scheme (ROHC) would lead to a bandwidth improvement for the video streams. In this thesis, for the performance evaluation purpose, we develop a pseudo ROHC function in our wireless emulator at Link Layer to show how the header compression scheme improves the bandwidth efficiency in our experiments. A detailed description of the ROHC scheme will be introduced in Section 4.4.

3.5.4.4 Traffic Shaping

Due to the highly bursty nature of compressed video streams, traffic shaping and traffic smoothing are required for efficient utilization of bandwidth and network resources at various points in a network. To transport Variable Bit Rate (VBR) video streams over Constant Bit Rate (CBR) CDMA channels, a traffic shaper at the source is required. In our wireless emulator, we use a transmission buffer as the simple leaky-bucket traffic shaper [42] at each transmitting end of wireless links at LL layer. Since the video stream is highly bursty, the peak data rate of the video bit stream is much higher than the bandwidth during some periods. The leaky-bucket traffic shaper reduces or removes the burstiness of the MPEG video stream, but also introduces large amounts of delay which can result in a large buffer requirement at the traffic shaper. Bursty video packets are fragmented into small PDUs first, and then are stored in the shaper buffer. Then they are transmitted with a given transmission rate, which is also the bandwidth of the wireless link.

3.5.4.5 Priority Scheduling Scheme

Queuing happens only when the transmission buffer is not empty. As long as the buffer is empty, new arrival packets will be transmitted without special treatment. Regular queues invariably employ the first in, first out (FIFO) principle: packets are transmitted in the same order they come in. When the queue is full, and additional packets come in, tail drops happen. The priority queuing scheme allows traffic to be classified in different priority. If there is any high-priority traffic, it's transmitted first. In our proposed retransmission-based ADSR-ARQ error control scheme, we introduce the priority scheduling to handle retransmission packets. In order to minimize the delay, the retransmitted packets are treated with higher priority than the first transmitted packets.

All of the above functions can be enabled or disabled by setting relevant parameters in Tcl scripts, which are used to modify the wireless network behavior for evaluations of relevant solutions. For streaming video, such applications may not require perfect reliability, but cannot tolerate high delay variation in data delivery as it causes a buffer depletion and discontinued playback.

In this Chapter, we first review conventional error control techniques in Section 4.1. Section 4.2 presents our proposed ADSR-ARQ in detail, including its design, components, service classes, as well as its analytical model and implementation. The buffer thresholds estimation of the ADSR-ARQ scheme is discussed in Section 4.3. Finally, the Robust Header Compression scheme (ROHC) is introduced in Section 4.4.

4.1 Conventional Transmission Error Control Techniques

The capacity of the wireless channel is fundamentally limited by the available bandwidth of the radio spectrum and various types of noise and interference. The resulting transmission errors require error control techniques. A classic technique is Forward Error Correction (FEC), but its effectiveness is limited due to widely varying error conditions. On the other hand, closed-loop error control techniques like Automatic Repeat Request (ARQ) are particularly attractive if the error conditions vary over a wide range, which has been shown to be more effective than FEC and successfully applied to wireless video transmission [43][44]. However, retransmission introduces variable delays in data delivery, which may not be acceptable for real-time applications.

4.1.1 Forward Error Correction

For wireless networks, FEC is necessary to reduce raw bit error rate. FEC is the simplest solution to improve the bit error rate seen by the higher layer protocol. However, it is inefficient when an error condition of the radio channel varies greatly. FEC adds parity bits to the transmitted packets. The receiver uses this redundancy to detect and correct errors. FEC can be applied on the physical layer, but is limited because it can only reverse a limited amount of

erroneous bits. FEC also wastes valuable bandwidth when correction is not necessary. However, FEC maintains constant throughput and has bounded time delay.

FEC can also be used at byte-level, which is done on a per-packet basis. Reed-Solomon (RS) codes [53] is one of the most widely used error correcting codes for packet level FEC. RS codes are linear block codes, which processes blocks of data at a time. Reed Solomon codes take a group of k data blocks and generate n-k FEC blocks. A receiver needs to receive any k of the n data or FEC blocks in order to recover the k data blocks. The RS algorithm can correct up to (n - k) / 2 symbol errors in each codeword. When the position of the symbol error is already known the algorithm is able to correct up to (n-k) erasures.

However, FEC incurs a constant transmission overhead even when the channel is loss free. Considering the limited bandwidth of the wireless link, it is important that the error control mechanism should be spectrally efficient.

4.1.2 Automatic Repeat Request

In contrast to FEC, ARQ requires a feedback channel from the receiver to the transmitter. ARQ only provides error detection capability by requesting retransmissions when the receiver detects errors. ARQ is simple, but its delay is variable and unbounded. There are three basic ARQ schemes in use: Stop And Wait (SW), Go Back N (GN), and Selective Repeat (SR) [45]. Though SR-ARQ requires buffering and reordering of out-of-sequence blocks, it provides the highest throughput. Packet loss recovery of ARQ is based on sequence numbers. The sender numbers all packets consecutively, and a missing packet is detected in the same way as an erroneous packet, either by the receiver's NACK or by the sender's time-out.

The ARQ solution is probably the closest to the one that will be implemented in 3G

cellular systems such as in UMTS with the RLC layer. It succeeds in reducing the observed bit error rate. However, in a wireless environment, fluctuations are dynamic and random, which cause burst data errors. The conventional SR-ARQ can solve the problem of data error, but with the cost of a long delay. The delay is caused by the persistent retransmissions. When using a conventional SR-ARQ, the sender and the receiver each has a window to keep track of data packet numbers. Under a deteriorated wireless channel, errors in both data and acknowledgement packets could prevent the sending and receiving windows from advancing. Consequently, the delays for the packets buffered at the receiver side will be notably long. Meanwhile the buffer at the sender side could overflow due to the high input data rate. The conventional SR-ARQ scheme also results in delay variation because of the randomness of error conditions. The timing requirement for video traffic will hardly be satisfied.

In order to overcome their individual drawbacks, the combination of these two basic classes of error control schemes, called hybrid FEC/ARQ schemes, have been developed. There are a couple of published papers [46][47] on performance evaluation of hybrid FEC/ARQ error control schemes. Their simulation results show that hybrid FEC/ARQ provides the best overall performance among the error control schemes. However, until today there is no widely accepted solution to the error correction problem of multimedia stream. The design of hybrid error control schemes is critically dependent on the application as different applications place vastly varying demands on the network. In this thesis, for the performance optimization, we propose and develop a modified ARQ error control scheme, called Adaptive Delay-constrained Selective Repeat ARQ.

4.2 Adaptive Delay-constrained Selective Repeat ARQ Scheme

It is well known that a wireless channel has burst errors, which generally cause either the loss of a whole packet or a large portion of the packet is corrupted. It is very difficult to use FEC only to recover such damaged packets with reasonable overhead and delay. On the other hand, ARQ is simple and achieves reasonable throughput levels if the error rates are not very large. However, in its simplest form, ARQ leads to variable delay that is not acceptable for real-time services. In a bandwidth and high bit error rate environment, it has been shown that the use of error-detecting codes and retransmission is better than the use of error-correcting codes [48]. Thus, all the lower layer protocols are primarily based on error detection and retransmission.

We propose an innovative Adaptive Delay-constrained Selective Repeat ARQ scheme (ADSR-ARQ) in radio link layer to optimize the performance of MPEG4 video streaming over wireless channels. The focus on the modified Selective Repeat ARQ scheme is motivated by its error recovery efficiency and simplicity of implementation. 3G UMTS cellular networks have incorporated the concept of Automatic Repeat Request (ARQ) in order to improve performance of data transfers over wireless [49]. In such bearers IP packets are segmented into small RLP PDUs at the sender and reassembled at the receiver. The ARQ mechanisms preserve reliability by re-transmitting RLP PDUs that are not received correctly. In addition, a feedback channel is available in our CDMA wireless emulator that simulates CDMA uplink and downlink two-way communication, thus a modified SR-ARQ scheme is developed to improve the performance of video streaming over wireless channels.

4.2.1 ADSR-ARQ Scheme Design

On error prone channels conventional ARQ schemes may introduce significant delays due to

the retransmission. These delays limit the applicability of conventional ARQ schemes to wireless multimedia communications. In order to overcome the drawbacks of conventional ARQ schemes, we propose an ADSR-ARQ scheme, which is capable of dynamically switching between ARQ and non-ARQ modes and adjusts the retransmission attempts based on the predefined transmission buffer length thresholds, to satisfy the overall delay constraint of video traffic. The objectives are to increase the throughput and reduce queue latency to bound realtime traffic delay. There is a tradeoff between packet loss and delay. For example, under critical error conditions, low-delay constraint may require that a corrupted packet should not be retransmitted. According to the video traffic and channel error conditions, ADSR-ARQ scheme defines three service policies with corresponding bounded retransmission attempts. The implementation incorporated a queue monitor, a packet scheduler and a timer manager to constitute the error control scheme.

4.2.1.1 Three Service Policies

Class 1 provides a timing bounded service with low probability of data loss. It delivers a data packet with limited numbers of retransmission to guarantee the delay bound. If the retransmissions are beyond the maximum allowed retransmission attempts, the sender will stop retransmission and drop that packet. Based on the experiments and observations, a maximum of two retransmission attempts are recommended.

Class 2 provides a timing bounded service with high probability of data loss. This policy has a tighter bound on retransmission attempts than *Class 1*. In this case, the maximum allowed retransmission is one.

Class 3 provides a timing bounded service without considering the data loss. This policy

bounds the delivery timing since there is no retransmission allowed in this case. That is, the ARQ mode is disabled.

These three service policies act only on the retransmitted packets. That is, whenever a retransmission is request, the ADSR-ARQ will make a decision which policy should be assigned to the retransmitted packet.

4.2.1.2 Components of ADSR-ARQ

In ADSR-ARQ, three major system components operate interactively to fulfill the service requirements, which are the queue monitor, the timer manager and the packet scheduler.

• Queue Monitor:

It monitors the queue length of the transmission buffer at sender, which provides the approximate estimation of video traffic and channel error condition. The more bursty the video traffic and the worse the error environment, the longer the queue in transmission buffer.

• Timer Manager:

It initiates a timer for each transmitted packet that is still in the retransmission buffer waiting for the receiver's acknowledgment, and also monitors a time-out event. When a time-out occurs, the corresponding packet is retransmitted automatically. In our research, we simplify this function in our implementation to just assign a timer to the significant packets, such as "control" packet etc.

• Packet scheduler:

This is the most important part in our ADSR-ARQ scheme. When the sender receives a retransmission request from the receiver, it determines the dispatching order of the packets

by using the "earliest deadline first" principle. The retransmitted packet has higher priority than the first transmitted one. For example, if the queue length reaches a certain threshold, this scheduler decides which packet should be served first according to the given priority and the assigned service policy. That is, the requested packet is dropped or retransmitted.

4.2.1.3 ADSR-ARQ Policy Based on Three-level Thresholds

We propose a mechanism that traces the length L of the transmission buffer and assigns different service policies we defined above to the retransmitted packets based on buffer length L. For instance, with ADSR-ARQ, whenever the sender receives a retransmission request, the packet scheduler of ASDR-ARQ will choose the requested packet from retransmission buffer and assign a service policy to this selected packet based on the current transmission buffer length. Herein, we define three-level buffer thresholds as Th1 and Th2 and Th3. Figure 4.1 illustrates ADSR-ARQ three-level buffer thresholds policy. This transmission policy works as follows:



Figure 4.1. ADSR-ARQ Three-level Buffer Thresholds Policy

1) *L*<=*Th1*: When the buffer occupancy is less and equal than Th1, Class 1 policy is assigned to the retransmitted packet. In this case, when the sender receives a retransmission request, based on the service policy, the requested packet is retransmitted twice at most.

2) Th1 < L <= Th2: When the buffer occupancy is greater than Th1, but less and equal than Th2, Class 2 policy is assigned to the retransmitted packet. In this case, when the sender receives a retransmission request, based on the service policy, the requested packet is retransmitted once at most.

3) Th2 < L <= Th3: When the buffer occupancy is greater than Th2, Class 3 policy is assigned to the retransmitted packet. In this case, when the sender receives a retransmission request, based on the service policy, no retransmission is allowed. That is, the requested packet is dropped. Further, if the length of the buffer reached Th3 then the new arriving packets are dropped based on drop-tail, where Th3 is the actual buffer size and cannot be exceeded.

The purpose of this policy is to constrain the queuing delay and retransmission delay. Low-delay constraint may require the corrupted packet to be dropped after exceeding its maximum allowed retransmissions. Thus there are packet errors at the decoder, which result in video quality degradation. There is always a tradeoff between packet loss and delay for realtime video applications. Video applications can be tolerant of certain packets errors. Part of the reason for this tolerance is due to the limits of human sensory perception. In many cases the requirements for error control and for end-to-end latency are contradictory. For real-time video transmission, delay is a more important performance issue than error rate.

4.2.2 Analytical Model

In order to have a clear understanding of what features the ADSR-ARQ scheme has and how it works, we developed an analytical model to describe its architecture and detailed implementation. In this thesis, we focus on the video streaming transfer taking place over the

45

downlink radio channel, where, with reference to the ADSR-ARQ scheme described above, the base station acts as sender and the mobile station acts as receiver. While constructing the ADSR-ARQ model, the following assumptions are introduced:

- The feedback channel (uplink) is error free, i.e. the acknowledgements (ACK or NACK) are never lost.
- 2) In our experimental model only video streams occupy the wireless channel, i.e. there is no other traffic mixed with the video stream in our research.
- 3) In this model, we assume there is a large enough buffer at the sender and receiver respectively. In our experiments, the window size is configurable by user. We set the window size as 30 packets at sender and receiver, which is large enough to avoid the case that the sending and receiving windows are consumed by waiting for the acknowledgements.
- 4) Both positive acknowledgement (ACK) and negative acknowledgement (NACK) are used in ADSR-ARQ scheme.



Figure 4.2. Radio Link Control (RLC) Layer Function Model with ADSR-ARQ Scheme

Figure 4.2 shows an RLC functional model with the ADSR-ARQ scheme in the 3G CDMA cellular networks. Unlike the traditional SR-ARQ design and implementation, ADSR-ARQ in the radio link layer has an asymmetric architecture for the sender and the receiver.

Only the sender has the functions of queue monitor, timer manager and packet scheduler. The receiver just has the functions of receiving data packets, buffering out-of-sequence packets and sending acknowledgements.

IP packets received from network layer are segmented into small RLP PDUs at the sender and reassembled at the receiver. At radio link layer, the sender keeps sending RLP PDUs from its transmission buffer to the receiver, and simultaneously storing the copies of these PDUs in the retransmission buffer waiting for the acknowledgements from the receiver. Upon receiving the expected PDU, the receiver sends back an ACK with next expected sequence number. Otherwise, the receiver buffers the out-of-sequence PDU and sends a NACK with the latest correctly received sequence number. ACKed PDUs are removed from the retransmission buffer. When the sender receives a NACK from the receiver, it means some PDUs are lost due to errors. The packet scheduler gets the feedback of transmission buffer occupancy from the queue monitor and makes a decision if the retransmission is allowed and assigns the retransmitted packet a proper service class described above. For high efficiency, we use cumulative ACK and NACK in the ADSR-ARQ implementation. In other words, upon receiving a cumulative NACK, the sender will resend all lost PDUs successively from the retransmission buffer instead of the next expected one in the conventional ARQ schemes.

4.2.3 ADSR-ARQ Implementation in The Simulation Model

In the ADSR-ARQ implementation, a retransmission flag in the link layer header is defined and the initial value is set to zero, which is used to count the retransmissions of the packet. When the sender receives a retransmission request, the ADSR-ARQ checks the transmission buffer occupancy and assigns a proper service policy to the requested PDU from the retransmission buffer. Then, the retransmission flag is checked to make a decision if the

47

PDU is allowed to retransmit or drop. If the PDU is retransmitted, the retransmission flag of the PDU automatically increases by one. Otherwise, the PDU is dropped, and, for the optimal design, those PDUs that belong to the same video packet are also dropped because they are now useless. Then, a "control signal" with higher priority is sent to the receiver to inform that certain PDUs have been dropped by the sender. The receiver immediately checks the receiving buffer to delete those PDUs which belong to the same video packets dropped by the sender and increases the ACK counter properly to send the buffered packets in sequence to upper layer from receiving buffer. Afterwards, a cumulative ACK is sent to the sender. In order to make sure that the "control signal" is not lost due to errors, the piggyback technique is used to let the following transmitted packets piggyback the "control signal" until the ACK is received. The "control signal" is also numbered in sequence. Therefore, the synchronization between sender and receiver is guaranteed. In our ADSR-ARQ, the retransmitted packets are assigned a high priority based on the earliest deadline first. Table 4.1 gives the RLC layer PDU header fields:

seqno_	bopno_	eopno_	rlctype_	pktno_	psize_	retxno_	ackno_	ctlno_
Table 4.1. RLC Laver PDU Header Fields								

seqno_: PDU sequence number. Theoretically this sequence number can wrap around. In our implementation, for the packet trace purpose, we use unique sequence numbers, which does not wrap around.

bopno : begin of packet number, i.e. the nth PDU of a packet

eopno_: end of packet number, i.e. the last PDU of a packet

rlctype : PDU type, e.g. DATA, ACK, NACK, or CTL ("control signal")

pktno_: packet number (from upper layer before fragment) which the PDU belongs
psize : PDU size

retxno : retransmission number

ackno : acknowledgement number

ctlno : piggy back for "control signal"



Figure 4.3. Simple Form of NACK-based ADSR-ARQ

In our implementation, any received packet in sequence is explicitly acknowledged as an ACK(n), where n is the next expected sequence number. Otherwise, the out-of-sequence packet is buffered and a cumulative NACK(n) indicates packet losses, where n represents the sequence number of the latest received packet. This implies that all unacknowledged packets whose sequence numbers are less than n are lost. Figure 4.3 gives a detail example of the operation of the cumulative NACK-based ADSR-ARQ scheme. For instance, PDU 0 and PDU 1 are sent and their acknowledgments are received. Due to burst errors, PDU 2, 3, 4 are lost. When the receiver receives PDU 5, since it is supposed to be waiting for PDU 2, the receiver buffers the out-of-sequence PDU 5 and then sends a negative acknowledgment NACK 5 to the sender. When the sender receives the NACK 5, that means unacknowledged PDU 2, 3, and 4 are lost. Thus the sender will resend the PDU 2, 3, 4 in sequence. Because of this successive retransmission for the burst packet losses, compared to the conventional SR-ARQ, the overall packet retransmission delay is obviously reduced.

4.3 Estimation of Buffer Length Thresholds in ADSR-ARQ

4.3.1 End-to-End Delay

Delays occur in the transmission queue due to the bursty video stream. The delay is highly dependent on the burst of the traffic through the link. Real-time applications have a firm delay bound. If a real-time data packet arrives late it is discarded. With the ARQ scheme in the link layer, retransmissions may result in many packets being dropped due to deadline expiry. In addition, a packet that remains in the sender's buffer whilst being late, may recursively produce additional delay to the following packets. The expired packets waste network resources and lead to long queuing delay for subsequent packets. In our ADSR-ARQ scheme, the overall delay is bounded based on three-level transmission buffer thresholds introduced above.

The end-to-end delay $T_{end-to-end}$ of a packet can be represented by the equation (4.1):

$$T_{end-to-end} = t_t + t_p + t_q + t_{arq}$$

$$(4.1)$$

Where t_t is the packet transmission delay, t_p is sum of the one-way wireless propagation delay and processing/scheduling delays, t_q is the queuing delay at the transmission queue, t_{arq} is the retransmission delay due to the packet loss. In our research the RLP PDU has a fixed size. From equation (4.1) we can see the first two parts $(t_t + t_p)$ are fixed to each packet. The delay jitter of a packet is mostly contributed by the queuing delay t_q and retransmission delay t_{arq} .

4.3.2 Three-level Buffer Length Thresholds

Delay jitter is a measure of variation in packet inter-arrival time at the receiver. The impact of jitter is reduced by a playout buffer at the receiver. This buffer allows the receiver to

play the stream in a continuous fashion even if some packets arrive late. However, delay beyond the size of the jitter buffer has the effect of the packet being lost. In this thesis, the playout buffer size of the reference software IM1 is measured in time unit. Since playout buffer size also means there is an initial playout delay of the video, the wait time before the audio/video starts to play is acceptable to the end users. For most commercial media player products, e.g. RealPlayer, Window Media Player, the playout buffer size is also measured in time unit, which can be usually configured by users from 5-15 seconds. Figure 4.4 shows the relationship between the playout buffer and delay/jitter. The left curve represents the transmission time of video data at the sender. The right curve is the video data playout time at the receiver. The middle solid curve is the video data reception time and the dotted one is the ideal reception time without queuing delay and retransmission delay. We can see that, as long as the video data arrives at the receiver before the playout deadline expiry.



Figure 4.4. Playout Buffer vs. Delay

Based on the relationship between delay jitter and playout buffer size, we use the following inequality (4.2) to approximately estimate the three thresholds (Th1, Th2, Th3) of transmission buffer described in Section 4.2.

$$t_q(n) + t_{arq}(n) \le B \tag{4.2}$$

Where B is the playout buffer size (seconds). $t_q(n)$ represents the queuing delay of the nth packet in the transmission buffer, $t_{arq}(n)$ is the maximum retransmission delay of the nth packet in terms of packet loss. Figure 4.5 depicts the three-level thresholds of transmission buffer.



Figure 4.5. Transmission Buffer Length Thresholds

The objective of the three-level buffer thresholds is to minimize video packet losses. Packet losses include packet errors in transport plus packets that arrive too late to be included in the jitter buffer for playout. Tradeoff between packet loss and delay is considered. For simplicity, we ignore those transmitted packets that are still queued in retransmission buffer waiting for the acknowledgements, since the retransmission buffer size is much smaller compared to the transmission buffer size. The following three scenarios are analyzed:

4.3.2.1 Scenario 1: (Th2<L<=Th3)

In this extreme case we assume that all PDUs transmit once with ARQ scheme disabled. The threshold Th3 of transmission buffer length, which guarantees expiration packet drop, is approximated as:

$$(Th3-1) \bullet \left(\frac{l}{R}\right) \le B \implies Th3 = \left\lceil \frac{B \bullet R}{l} + 1 \right\rceil$$
 (4.3)

Where l is the RLC layer fragment PDU size, R represents the transmission rate. That is, when a new packet arrives at the transmission queue, if the observed queue length is longer than Th3, then the packet will be dropped by the receiver due to its deadline expiry even if the wireless channel is clean with no packet retransmission. A queue capacity larger than Th3 is clearly not useful.

4.3.2.2 Scenario 2: (Th1<L<=Th2)

Due to the noisy wireless channels, PDUs might be lost and need to be retransmitted. Under an error probability p, in order to satisfy the video quality as well as meet the delay constraint of the video stream, for generality, we assume that the maximum allowed retransmission attempt is n. From a statistic point of view, there are an average of p percent of PDUs in the transmission buffer that might encounter errors and have to be retransmitted up to n times (the worst case). From equation (4.2), we have:

$$(1+np)\bullet(Th2-1)\bullet\left(\frac{l}{R}\right)+n\bullet RTT \le B \implies Th2 = \left\lceil \frac{(B-n\bullet RTT)}{(1+np)}\bullet\left(\frac{R}{l}\right)+1 \right\rceil$$
(4.4)

Where RTT is the round trip time from sender to receiver. In our model, we assume RTT is much less than the average queuing delay. When a new packet arrives at the transmission queue, if the observed queue length is longer than Th2, the ADSR-ARQ scheme should disable the retransmission and drop the requested packets from the retransmission buffer. Otherwise the new arrival packets will have expiration risks because the retransmissions of those packets ahead could lead to a long queuing delay.

4.3.2.3 Scenario 3: (L<=Th1)

The difference between scenario 2 and scenario 3 is the maximum allowed retransmission attempts. In this case we assume that the maximum retransmission attempt is n', which is larger than n. That means the lost packets are allowed to retransmit as much as possible to recover the transmission error when less packets are in the transmission buffer.

$$Th1 = \left\lceil \frac{(B - n' \bullet RTT)}{(1 + n'p)} \bullet \left(\frac{R}{l}\right) + 1 \right\rceil$$
(4.5)

The above derivations of Th2 and Th1 are statistic results. For extreme cases, we can assume p=1, which gives the tightest bound. That is, all the packets ahead happen to encounter errors, which is not likely in reality though. To achieve the optimum results, it is quite clear that the selection of this threshold depends entirely on the limitations of the real-time session and the round-trip time between the sender and the receiver, as well as playback buffer size. In practice, it is hard to define a precise threshold. In our experiments, the above n and n' are set to 1 and 2 respectively.

4.4 Robust Header Compression (ROHC) Scheme

Robust header compression scheme supports efficient use of scarce wireless resources for multimedia data with real-time constraints [41]. Wireless networks of the 3rd generation (3G) will offer a wide range of Internet Protocol (IP) based multimedia applications [50], which require more bandwidth and are highly delay sensitive. Multimedia applications often use Real Time Protocol (RTP), User Datagram Protocol (UDP) and IP. Each of the protocol layers adds a significant header overhead. While the compression of multimedia payload is mostly sufficient or even excellent, the most promising compression gain can be yielded by focusing on the RTP/UDP/IP header. IP header compression mechanisms have always been an important part of saving bandwidth over bandwidth-limited links. For multimedia services in wireless environments Robust Header Compression (ROHC) [51] was introduced. Robust header compression was standardized by the Internet Engineering Task Force (IETF) in RFC 3095 and will be an integral part of release 4 and 5 of the 3GPP-UMTS specifications. This compression scheme was designed to operate in error-prone environments by providing error detection and correction mechanisms in combination with the robustness for IP based data streams.

ROHC in its original specification as in RFC 3095 is a header compression scheme with profiles for three protocol suites: RTP/UDP/IP, UDP/IP and ESP/IP. In Figure 4.6 the combined header for a real-time multimedia stream with IPv4 is given. This includes the 20 bytes IPv4 header, the 8 bytes UDP header and the 12 bytes RTP header. Redundancy exists among the different headers, in particular between consecutive packets belonging to the same IP flow.



Figure 4.6. Protocol Stack and Header Structure in ROHC

As shown in Figure 4.6, ROHC is located in the standard protocol stack between the IP network layer and the link layer. The MPEG-4 streaming server in our wireless multimedia testbed was implemented in UDP/IP suite, which is covered by ROHC scheme. Therefore, to evaluate the performance of MPEG-4 video streaming over CDMA wireless testbed, the ROHC scheme is introduced in our wireless emulator, which is depicted in Section 3.5. In this thesis, we focus on the performance evaluation of ROHC, rather than the implementation of ROHC scheme, which exceeds the scope of this thesis. Relevant information about ROHC can be seen in [41][51].

IP based header compression is based on the fact that headers have significant redundancy. In wireless communication, the IP overhead (including the RTP and UDP headers) sometimes significantly decrease the efficient throughput. The overhead ratio η in a packet is given in equation (4.6):

$$\eta = 1 - \frac{Payload}{Header + Payload} = \frac{Header}{Header + Payload}$$
(4.6)

The overhead ratio depends only on the packet length. Video streams are characterized by variable packet length. Obviously, the smaller the IP packet, the higher the impact of the IP overhead. By using the ROHC, for the smaller video packets, the bandwidth saving gain is significant. The size of the video frames depends on the content of the video sequence and the encoder settings used. Also the video packets' size depends on the streaming technologies.

Chapter 5: Performance Evaluation

In this chapter we present the performance evaluation of MPEG-4 video streaming over wireless networks. First, we describe the methodology of simulations including the testbed setup, measurement methods and the assumptions and limitations in our experiments in Section 5.1. Then, Section 5.2 shows the experiment results, including the channel model verification tests, ROHC scheme bandwidth efficiency tests, and our proposed ADSR-ARQ error control scheme performance evaluation compared to the conventional SR-ARQ scheme. We conducted experiments to show that ADSR-ARQ scheme is both a feasible and beneficial means of improving the quality of received video. Finally, a summary is presented in Section 5.3.

5.1 Methodology

5.1.1 Experiment Goals

In this thesis, we evaluate the suitability of the current networking technologies available for the transmission of MPEG-4 video streaming over wireless links. Additionally we investigate possible improvements for the performance of MPEG-4 video stream transmissions under different wireless network conditions. The goals of the measurements are:

- To verify the developed burst channel model in our wireless emulator
- To evaluate the ROHC header compression scheme performance
- To investigate how the bandwidth affects the video streaming transmission
- To evaluate the ADSR-ARQ scheme performance on delay, throughput and video quality versus packet error/loss.

• To study quantitatively the tradeoff between packet loss and delay in terms of the QoS of video quality

5.1.2 Experiment Setup

In order to perform the experiments, a controlled network environment is needed for maintaining consistent input parameters and testing conditions. In this thesis, an MPEG-4 video streaming over CDMA wireless network testbed is proposed and built. Since it is widely accepted that future mobile communication infrastructures will be all IP-based, this allows us to use our office LAN to study the behavior of wireless streaming media applications. By incorporating our deployed wireless emulator and the open source MPEG-4 streaming server and client, which were introduced in Section 3.5 and Section 2.4 respectively, a LAN-based real time wireless video streaming testbed is set up, which allows us to investigate the behavior of video streaming applications under various network conditions.



Figure 5.1. LAN-based Emulated Wireless Video Streaming Testbed

This wireless video streaming testbed is built upon 3 PCs, shown in Figure 5.1. Each of the system components, i.e. the MPEG-4 streaming server, the wireless emulator and the client, runs on an individual PC respectively. The testbed consists of some hardware components and associated software and network protocols. All components are located within an isolated Ethernet to make sure no packet losses occur in the wired portion of the network.
5.1.2.1 MPEG-4 Streaming Server

The streaming server is a Pentium II PC running Windows 2000. The open source Apadana Server is installed and compiled under VC++ 6.0 to generate the executable streaming server, which reads the local pre-recorded mp4 file upon request from the client and streams it to the client through the network. The detailed implementation of the Apadana Server can be found in [26][27][28]. Some related information was introduced in Section 2.4.

5.1.2.2 Wireless Emulator

The wireless emulator is a Pentium II PC running FreeBSD and Network Simulator 2 (NS2) software. Detail configuration and implementation of the wireless emulator were described in Section 3.5. This PC has 2 network interface cards (NIC) which are connected to the different IP subnets. The wireless network emulator introduces wireless characteristics such as delay and packet error/loss to traversing data packets. In terms of the source and destination IP addresses, the emulation programs capture packets from one network interface card and inject them to the other network interface card after processing. The emulator performs duplex transportation. For example, the down-link (from base station to mobile station) channel mainly transports the video streaming traffics and control signals from server to client, whereas the up-link (from mobile station to base station) channel transports requests, control signals and acknowledgements from client to server.

5.1.2.3 Client Terminal

The Client terminal is a Pentium II PC running Windows 2000 and IM1 player (MPEG-4 reference software), which sends the request to streaming server based on the server's IP address and video filename and decodes the received video streams, then displays them on the PC screen. For the purpose of an analysis, we modify the IM1 player (open source) source

codes to capture the received video frames and write them into an RGB file.

5.1.3 Measurement Methods

As origin video source, two video signals with the Quarter Common Intermediate Format (QCIF) (176x144 pixels) that were stored in mp4 files were used, which were produced to test the performance of MPEG-4 streaming server in [26][27][28]. The QCIF format is the common video size on a mobile device. One video clip called "Suzie.mp4" has 150 frames and a frame rate of 15 frames per second, which average bit rate is around 20Kbps. Another video clip named "Video.mp4" has 587 frames with a frame rate of 15 frames per second, which has around an average bit rate of 25Kbps. These two pre-recorded video clips provide variable bit rate (VBR) traffic streams, which result in packet bursts.

All tests are performed on the real-time wireless video streaming testbed. During the experiment run, the MPEG-4 streaming server and wireless emulator are always on, the client launches IM1 Player and inputs the server's IP address and video filename at open URL option, as well as displays the received video stream on the screen. The wireless emulator can simulate IS-95 and CDMA2000 systems by given a proper bandwidth and emulated delay and packet loss. All the developed functions and test parameters can be set through NS2 Tcl scripts by the user, such as the channel bandwidth (transmission rate or slot timer), channel error models selection, packet error rate, burst length, ROHC header compression scheme ON or OFF, as well as error recovery scheme selection, etc.

In order to analyze the performance, all real video packets are traced within the wireless emulator. The video packets are assigned sequence numbers and time-stamped using the NS2 real-time scheduler when they enter into and leave the wireless emulator. The timer starts from the first packet. The packet loss or retransmission behavior is recorded as well. More than 100 timed tests under different wireless conditions and video clips have been done over this testbed. From our visual observation and experiment data analysis, under the same scenario, the test results were stable and repeatable. In this thesis, the measurements were done in the following manner:

- Every measurement run created different groups of results for comparison, e.g. different bandwidth comparison, non-ROHC vs. ROHC, ADSR-ARQ vs. SR-ARQ.
- During any particular run, there is no interaction between client and server until the video stream transmission is over.
- For every group of results, 5-10 repeated measurements have been done, so that the results have a statistical meaning.
- From each run, the average values or the representative results were used for presentation.

5.1.4 Experiment Assumptions

Due to the technical limitations on our wireless emulator and real-time testbed, to simplify the test, we assume:

- Due to the NS2 emulation mode limitation, the wireless emulator does not support mobility and routing at this stage. There is just one base station and one mobile station in the emulator.
- 2) In this thesis, we focus on the wireless domain, so the Internet delay is ignored. That is, in our experiments, the end-to-end delay is measured starting from the video packet entering into the wireless channel to leaving it. During an experiment run, the bandwidth of wireless channel is not fluctuated with time.
- 3) In our experiments, the round trip time (RTT) in the wireless channel is around 40ms

and the processing delay and propagation delay are small and ignored compared with queuing delay.

- Our channel models use a fixed average packet error rate (PER), that is, the average PER is not varied with time.
- 5) To focus the performance evaluations on the video stream transmission, we make the simplifying assumption that the feedback channel (uplink) is error-free.
- 6) In our experiments, our proposed ADSR-ARQ only demonstrates its validity in realtime traffic at the current research stage to overcome the conventional SR-ARQ long delay drawback in real-time applications. For simplicity, just a single video traffic occupies this emulated wireless channel in the test.
- 7) To compare the performance of ADSR-ARQ with the conventional SR-ARQ, we use the SR-ARQ scheme in the contributed GPRS module in NS2 as the conventional SR-ARQ, which was implemented by *Richa Jain* at IITB, India [52]. This SR-ARQ scheme has also been used in other relevant research [40].
- 8) ROHC scheme is implemented as a pseudo function in the wireless emulator, the compression ratio is assumed to be a fixed value from the statistic point of view. And the processing delay of ROHC is ignored in our experiments.

5.2 **Experiment Results**

The experiments were done over the wireless video streaming testbed. The experiments included three parts. In the first part, the test was to verify the error model to show wireless error characteristics. Next, the performance of ROHC header compression scheme was evaluated to show its bandwidth efficiency. Finally, we evaluated our proposed ADSR-ARQ error control scheme to show its performance improvement compared to the conventional SR-

ARQ.

5.2.1 Channel Error Model verification

5.2.1.1 Packet Error Rate

Error-prone is one of the most important characteristics in wireless networks. Error rate is the significant parameter for wireless networks. This parameter can be defined in a number of ways, such as bit error rate, packet error rate, or frame error rate. Packet loss in the wireless domain is instigated by channel errors. In our wireless emulator, we employed packet-level error models rather than bit error models to study how random packet loss and burst packet loss affect the performance of video streaming transmission. In our experiments, we use packet error rate (PER) to indicate the channel error conditions, which is defined as the ratio of the average number of corrupted or error packets to the total number of packets that are transmitted. In this thesis, unless otherwise stated, the term "packet" in PER specifically represents the RLC layer fragmented PDU in our wireless emulator (see Section 4.3). To compare PER with BER, we use the following formulation to derive BER from PER, where L is the packet length.

$$PER = 1 - (1 - BER)^{L} \implies BER = 1 - (1 - PER)^{1/L}$$
(5.1)

Equation (5.1) is valid for uncorrelated bit errors. For the burst bit errors case, although many researchers were involved in this research area, it is hard to find an exactly simple expression for packet error rate, where the success of a packet depends jointly on the errors on the various symbols [54]. Herein, Table 5.1 shows the comparison between PER and BER by given different fragmented PDU size based on equation (5.1), which represents the random bit error conditions. For the burst bit error conditions, the PER is less than the result of equation (5.1). In our experiments, the burst PER ranging from 1% to 15% approximately equals BER ranging from 10^{-5} to 10^{-3} , which represents the typical error conditions in real wireless network.

PDU size = 24 Byte		PDU size = 96 Byte		PDU size = 304 Byte	
PER	BER	PER	BER	PER	BER
1%	5.23x10 ⁻⁵	1%	1.28x10 ⁻⁵	1%	4.13x10 ⁻⁶
5%	2.67x10 ⁻⁴	5%	6.54x10 ⁻⁵	5%	2.11x10 ⁻⁵
10%	5.49x10 ⁻⁴	10%	1.34x10 ⁻⁴	10%	4.33×10^{-5}
15%	8.46x10 ⁻⁴	15%	2.07x10 ⁻⁴	15%	6.68x10 ⁻⁵

Table 5.1. Packet Error Rate vs. Bit Error Rate

5.2.1.2 Error Pattern

To verify if the error models can provide the wireless error characteristics we desired, the experiments are performed to show the random packet error pattern and burst packet error pattern in Figure 5.2 and Figure 5.3. In the test, we streamed a 60sec long video clip "Video.mp4" over the wireless multimedia testbed. The burst error model generates the error patterns based on the two parameters of average PER and average error burst length. The corrupt video packets are flagged and traced in the wireless emulator.

Figure 5.2 shows the error patterns with different average error burst lengths given an average 10% PER. We can see that the random error model generates a uniform error distribution pattern with time, whereas, with the increase of burst length, the burst error distribution becomes more discrete and the total number of error packets decreases slightly. The less the burst length, the closer to the random uniform distribution. This observation is in agreement with the expected wireless error characteristics. Figure 5.3 shows the error patterns with different average PER given an average error burst length 7. We can see that the slope of the error pattern increases with the PER increase. That is, the total number of error packets over a given time period increases with the PER.



Figure 5.2. Packet Error Pattern vs. Average Error Burst Length



Random PER=4%
 Burst PER=6%
 Burst PER=8%
 Burst PER=10%

Figure 5.3. Packet Error Pattern vs. Average Packet Error Rate

This test shows that our developed error models closely reflect the error characteristics in real wireless networks. By using these error models, investigating or verifying other new algorithms and solutions over wireless channel is feasible and valuable.

5.2.2 ROHC Performance Evaluation

Robust Header Compression (ROHC) has recently been proposed to reduce the large protocol header overhead when transmitting multimedia over RTP/UDP/IP in wireless networks. The ROHC scheme is implemented as a pseudo function and integrated into LL of the wireless emulator. Our performance evaluation of ROHC is from the network providers' point of view, and focuses on the efficiency of the header compression. From the published relevant research [41], the average header compression ratio of ROHC over RTP/UDP/IP can reach around 84%. In our experiments, we use this compression ratio to evaluate the end-to-end delay and bandwidth efficiency. Since our MPEG-4 streaming server (Apadana Server) uses UDP/IP to stream video packets, the overhead saving of a video packet with ROHC is around 24Bytes. In the test, video clip "Suzie.mp4" is used and the wireless channel is error free. A small PDU size (24Byte) and low bit rate bandwidth (9.6K/19.2K) are chosen for this test. Every measurement run creates two groups of results: One with ROHC and one without ROHC. So the comparison can easily show the bandwidth saving by using ROHC header compression.

Figure 5.4 displays the end-to-end packet delay comparison between ROHC and non-ROHC with bandwidths of 9.6Kbps and 19.2Kbps, respectively. From the curves, we can see that the ROHC significantly decreases the end-to-end delay. For bandwidths of 9.6Kbps and 19.2Kbps, ROHC decreases delays by 17.5% and 33.6%, respectively, compared to the cases without ROHC. Obviously, using ROHC effectively increases the throughput, as seen in Figure

5.5. For bandwidths of 9.6Kbps and 19.2Kbps, the throughput with ROHC achieves 11.6% and 13.3% gain over cases without ROHC, respectively.



Figure 5.4. Error Free Case: End-to-End Packet Delay with ROHC



Figure 5.5. Error Free Case: Throughput with ROHC

Figure 5.6 and Figure 5.7 show the end-to-end packet delay and throughput comparison

between ROHC and non-ROHC with bandwidth of 9.6 Kbps under burst error environment. We can see that ROHC decreases delay by 14% and improves throughput by 10% on average, compared to the cases without ROHC.



Figure 5.6. Burst Error Case: End-to-End Packet Delay with ROHC



Figure 5.7. Burst Error Case: Throughput with ROHC

The above results verify that ROHC header compression can dramatically improve the bandwidth efficiency by reducing the protocol header overhead. For the bandwidth limited wireless network, especially for the low bit rate, e.g. 9.6Kbps of IS95 capacity, such bandwidth saving is more significant. ROHC header compression is definitely an efficient solution for the performance improvement of wireless multimedia transmission.

5.2.3 Influence of Network Bandwidth

From the network's point of view, the bit rate of a network (also called bandwidth) is a crucial network characteristic. An acceptable level of quality-of-service (QoS) for the transmitted video stream will be provided if the wireless network can guarantee a minimum bandwidth for the encoded video stream. For a real-time application, an ideal bandwidth would be the peak rate. If a bandwidth equal to the peak rate is allocated to the video stream, each packet will arrive at the receiver early enough so that it can be played back in its proper time instant. On the other hand, if the allocated bandwidth is smaller than the peak rate, some packets will not arrive in time for playback. In such cases, a buffer will be used to store the packets. The packets will be played back after an appropriate amount of traffic is accumulated in the buffer. The packets will be collected in the playback buffer and will be read into the decoder in a constant rate. When the available bandwidth is extremely less than the average media bit rate, the quality of streaming applications is not acceptable without a big playback buffer.

For the given pre-recorded video file "Suzie.mp4", the test shows how the channel bandwidth affects the video transmission. Figure 5.8 gives the relationship between bandwidth and end-to-end packet delay of video streams transmission. We can see the end-to-end packet delay of the video stream decreases dramatically with the bandwidth increase. In this test, with

9.6 Kbps channel bandwidth, the maximum packet delay can reach 20 seconds, which is much greater than the 5 seconds delay over 19.2 Kbps channel bandwidth. In the low bit rate bandwidth, the end-to-end bandwidth mainly contributed by queuing delay when the round trip time (RTT) and propagation delay are relatively smaller. When the bandwidth is large enough, say, much greater than the video average bit rate, in this case with 38.4 Kbps, the end-to-end delay is less than a hundred milliseconds since there is almost no packets queued.



Figure 5.8. End-to-End Delay vs. Bandwidth



Figure 5.9. Throughput vs. Bandwidth

Figure 5.9 shows the throughput performance of video stream with different bandwidth. The larger the bandwidth, the higher throughput the video stream gets. From the curves in Figure 5.9, the throughput with bandwidths 9.6Kbps and 19.2Kbps increases fast at the first 10 seconds, then, the throughput gradually achieves the maximum capacity, whereas, the shape of the curve with bandwidth 38.4Kbps is different than its counterparts of 9.6Kbps and 19.2bps. That is because there is no queuing delay due to the large bandwidth. Therefore, its throughput curve represents the encoded video bit rate approximately.

In the experiments, the queuing delay is the major factor resulting in a long end-to-end delay. In addition, from our observations, when the bandwidth decreases to 9.6 Kbps, which is also the capacity of current 2G IS95 cellular systems, the video playback obviously was delayed for several seconds and the video quality was degraded due to the long delay. However, it still shows that streaming MPEG-4 video over a low bit rate wireless channel is possible. We also tested another video clip "Video.mp4" with 25 Kbps average bit rate. The IM1 player was not able to play the video clip under 9.6Kbps bandwidth since the required minimum bandwidth was not available for this video transmission.

The conclusion of this test is that the bandwidth of networks greatly affects the video stream transmission quality. A remarkable note of this test is the fact that it is actually possible to stream a video file through a low bandwidth wireless network. To guarantee the QoS of video quality, in general, the acceptable bandwidth of the network has to match the average media bit rate at the minimum. In other words, a minimum bandwidth is required to stream video over the networks, which depends on the encoded video bit rate and playback frame rate.

5.2.4 ADSR-ARQ Error Control Scheme Evaluation

In this section, we investigate ADSR-ARQ performance over our wireless video streaming testbed. The main purpose of ADSR-ARQ error control scheme is to overcome drawback of long delay of the conventional SR-ARQ scheme due to persistent retransmissions, by trading off the packet error and delay to minimize the packet loss and meet the overall QoS requirements of a video stream. To evaluate the ADSR-ARQ scheme, end-to-end packet delay, delay jitter, throughput, and video quality versus packet loss are measured and compared to the conventional SR-ARQ scheme. The measurements were done in this manner, i.e., every run created two groups of measurement results: One for the conventional SR-ARQ and one for the ADSR-ARQ.

In the experiments, we use equations (4.3), (4.4), (4.5) derived in Section 4.3.2 to estimate the three-level buffer length thresholds of ADSR-ARQ. The experiment parameters for the wireless link, ADSR-ARQ scheme and real-time application are given in Table 5.2.

Parameters	Value
Wireless transmission rate	54 Kbps
RLP PDU fragment size	135 Byte
Round trip time (RTT)	40 ms
Average packet error rate (PER)	10%
Average error burst length	7 PDUs
Max. playout buffer size	15 s
Test video clip	"Video.mp4"
Transmission buffer threshold of ADSR-ARQ Th3	750 PDUs
Transmission buffer threshold of ADSR-ARQ Th2	680 PDUs
Transmission buffer threshold of ADSR-ARQ Th1	622 PDUs

Table 5.2. Experiment Parameters

5.2.4.1 Delay Jitter

Jitter is a measure of variation in packet inter-arrival time. Delay jitter is the main enemy of streaming video since it may cause data packets to be received late and became useless and might have "snowball" effects on the following packets. The jitter is mainly due to the delay through transmission over the network. In particular, in the presence of packet losses, packet retransmissions introduce significant jitters. Although a playout buffer at the receiver side is usually used to compensate for the jitter, the large jitter that the buffer cannot handle still degrades the video quality due to the playback deadline expiry. In designing a multimedia network, it is important to place an upper limit on the permissible jitter. In our experiments, the delay jitter is calculated as the variation of packets delay at receiver.





Figure 5.10. Delay Jitter Comparison: (a)Non-ARQ (b)SR-ARQ (c)ADSR-ARQ

Figure 5.10 shows the comparison of the end-to-end packet delay jitter over 2050 video packets between ADSR-ARQ and SR-ARQ. Without ARQ scheme, shown in Figure 5.10(a), the delay jitter is less than 100 milliseconds. However, due to the retransmission of error packets, the SR-ARQ scheme results a large delay jitter, shown in Figure 5.10(b). We can see the maximum delay jitter in this case is around 2.8 seconds. This large delay jitter cannot satisfy the QoS requirements of video streaming applications. On the other hand, from Figure 5.10(c), we can see that the ADSR-ARQ scheme reduces the delay jitter dramatically compared to the SR-ARQ scheme. Since ADSR-ARQ scheme uses limited retransmission attempts based on transmission buffer length, and successive retransmission of lost packet bursts based on the cumulative NACK, as well as priority scheduling design for retransmitted packets, the delay jitter is bounded within an acceptable level.

The test demonstrates that our proposed ADSR-ARQ scheme significantly reduces the delay jitter by minimizing the packet loss due to long delay or transmission error. It is obvious that the ADSR-ARQ is more suitable for video stream applications than the SR-ARQ.

5.2.4.2 End-to-End Delay and Throughput

ADSR-ARQ error control scheme aims to minimize the packet losses to improve the overall video quality. There is a trade-off between the packet loss and delay. Since a multimedia application is loss-tolerant up to a certain degree but it is more sensitive to delay, control of packet loss and delay can be used to improve the perceived quality. In our experiments, the end-to-end packet delay is measured from the time of the packet entering into the wireless network to the time of the packet leaving the wireless network. The throughput of a network is its effective bandwidth. Herein the average throughput is measured as the successfully received data bits during a cumulative time period at the receiver.



Figure 5.11. ADSR-ARQ vs. SR-ARQ: (a)End-to-End Delay (b)Queue Length

Figure 5.11(a) shows the comparison on end-to-end packet delay between ADSR-ARQ and SR-ARQ scheme. As we expected, the overall end-to-end packet delay with ADSR-ARQ is significantly reduced compared to the conventional SR-ARO. In this case, the ADSR-ARO can yield 60% improvement on the average delay. In addition, the packet delay with SR-ARQ increases at a faster rate than that with ADSR-ARQ due to the queuing delay and unbounded retransmission delay. We also noticed that the packet delay curve with SR-ARQ has larger fluctuations over time. That is because whenever there is a retransmission due to packet error the packet delay increases greatly which affects the following packets as well, whereas, the ADSR-ARQ scheme has bounded retransmissions based on transmission queue length and an optimal design with cumulative NACK, which significantly reduces the queuing delay and retransmission delay. The trade-off of ADSR-ARQ is that some packets are dropped under heavy overload to guarantee overall performance so that the overall video quality is still acceptable. Figure 5.11(b) depicts the transmission buffer status with time. We notice that the shape of the curves in Figure 5.11(a) is similar as in Figure 5.11(b), which also confirms that the queuing delay is the main contributor to end-to-end packet delay in our experiments when the round trip time (RTT) is small. And there is a big fluctuation caused by the media traffic on the extreme right of each curve. That is because there is a suddenly fast motion in the last part of the video, which results in a packet burst. In Figure 5.11(b), we just monitored the queue length until no new video packets were coming.

Figure 5.12 shows the throughput comparison between ADSR-ARQ and SR-ARQ. We can see that ADSR-ARQ has improved the overall throughput by about 35% on average compared with SR-ARQ because of the lower end-to-end packet delay of ADSR-ARQ. We also notice that the curve of SR-ARQ exhibits large fluctuations due to the retransmissions. The throughput without ARQ apparently is higher than that with ARQ since retransmissions

decrease the throughput.



Figure 5.12. ADSR-ARQ vs. SR-ARQ: Throughput

In order to show how the ADSR-ARQ scheme performs better than the conventional SR-ARQ scheme, we compare these two schemes on end-to-end packet delay and throughput by varying different parameters. Figure 5.13 shows the comparison between ADSR-ARQ and SR-ARQ under different packet error rates. We varied the mean packet error rate from 0 to 20% and fixed the mean burst length to 7 packets. We can see, in Figure 5.13(a), with the increase of the packet error rate, the end-to-end delay with SR-ARQ increases dramatically compared with ADSR-ARQ. The higher the PER, the longer the delay the SR-ARQ produces compared to ADSR-ARQ. The end-to-end packet delay with ADSR-ARQ is bounded and is much less than with SR-ARQ given a high PER. On the contrary, the overall channel throughput with SR-ARQ decreases much more than with ADSR-ARQ when the packet error rate increases to 15%, the SR-ARQ cannot work properly. Actually the IM1 player cannot play the video due to the long delay. Therefore, we are unable to compare those data in these failure cases in Figure 5.11. This test indicates that ADSR-ARQ outperforms the conventional SR-ARQ under





Figure 5.13. ADSR-ARQ vs. SR-ARQ: (a)Delay and (b)Throughput vs. PER

In our live video tests, it is not easy to trace the late packet dropped by the decoder because the MPEG-4 reference software, IM1 player, does not explicitly provide this function, The coding of IM1 player exceeds the scope of this thesis, which results in the limitations of our experiments. We only can calculate the dropped frames by capturing the displayed images

on the screen. However, from our observation, the long delay with conventional SR-ARQ results in the degradation of video quality.



Figure 5.14. ADSR-ARQ vs. SR-ARQ: (a)Delay and (b)Throughput vs. Bandwidth

Figure 5.14 shows the comparison between ADSR-ARQ and SR-ARQ under different channel bandwidth. We measure the performance with 54K, 64K, 96K and 144Kbps channel bandwidth under the same error condition. The curve of ADSR-ARQ in Figure 5.14(a) has the

similar trend as the one of SR-ARQ. In this case, ADSR-ARQ reduces end-to-end packet delay by 73% on average over SR-ARQ. Similarly, Figure 5.14(b) shows that ADSR-ARQ yields 37.3% gains in throughput on average over SR-ARQ. This test shows that ADSR-ARQ has overall better performance than SR-ARQ with different channel bandwidths.





Figure 5.15 shows the comparison between ADSR-ARQ and SR-ARQ under different error burst length. From Figure 5.15(a), with the increase of error burst length, the curve of ADSR-ARQ remains flat, whereas, the end-to-end delay with SR-ARQ increases with the burst length. That means the error burst length affects the performance of SR-ARQ much more than ADSR-ARQ. Because of the optimal design of successive retransmission of bursty lost packets in ADSR-ARQ, the error burst length doesn't affect its performance too much in our experiments. Obviously, the throughput has corresponding results in Figure 5.15(b).

From above test results, it is apparent that ADSR-ARQ scheme yields great improvements on the overall throughput and end-to-end delay. As we expected, ADSR-ARQ gives a better performance than conventional SR-ARQ under different wireless channel conditions. We also can see that the conventional SR-ARQ is not suitable for video streaming application due to the long delay and the delay jitter. The fact behind the long delay with conventional SR-ARQ is that the delay propagates and finally results in packets expiry. From our observations, this type of accumulated packet delay in conventional SR-ARQ makes the video freeze sometimes and even stop to play from a certain point because the late arrival of packets are dropped by the decoder due to deadline expiry.

5.2.4.3 Packet Loss vs. Video Quality

The purpose of this measurement is to show how the received video quality is affected by packet losses. Packet losses include packets actually lost in transport plus packets that arrive too late to be included in the jitter buffer for display on the user's screen. Generally, the quality of a video is a subject of personal opinion, so this means that the quality of service improvement for a video transmission has the only goal to satisfy the average human being watching the contents of the video. In order to analyze live video quality, some amount of

subjective evaluation would be necessary. To see the effect of packet drops on the quality of presentation, some snapshots of the rendered video can be shown.

Packet Error	Withou	it ARQ	With ADSR-ARQ	
Rate	Video Frame Drop	Frame Error Rate	Video Frame Drop	Frame Error Rate
5%	7	4.7%	0	0
10%	15	10%	0	0
15%	25	16.7%	1	0.7%

l adle 5.3. Packet Error Rate vs. video Frame Los



(a) No Error



(b) PER = 5%

(without ARQ)



(c) PER = 10% (without ARQ)



(d) PER = 10% (with ADSR-ARQ)

Figure 5.16. Snapshots of Frame 65 in the "Suzie" Sequence

The "Suzie.mp4" file is used in the test, which has total 150 frames. The fragment size in RLP is 96 Bytes and the error burst length is 5. Since some video packets dropped due to transmission errors, some video frames could not correctly be decoded in decoder, resulting in the video frame drops. Table 5.3 depicts the relationship of packet error rate vs. video frame loss. In the case without ARQ, we notice the ultimate frame error rate approximates the packet error rate. Obviously, ADSR-ARQ significantly reduces packet losses. Furthermore, from our observations, the image quality also becomes gradually worse with increasing packet error rate. Figure 5.16 gives the snapshots of frame 65, which shows how increasing the packet error rate greatly degrades the quality of the received image. Image 5.16(a) is the original. Image 5.16(b) is blurry due to some packet losses. Because the first I-frame was lost, image 5.16(c) almost cannot be seen clearly. Image 5.16(d) is with ADSR-ARQ, which provides a good video

quality by recovering packet errors. The test is performed to show how the packet error/loss affects the video quality and our ADSR-ARQ error control scheme can significantly improve the overall video quality since the packet losses are reduced by ADSR-ARQ.

For a quantitative analysis of video quality in terms of transmission over wireless networks, we need a metric that compares the reconstructed frame at the receiver side with the original frame. In practice, MSE (Mean Squared Error) and PSNR (Peak Signal-to-Noise Ratio) are the most common distortion measures for objective video quality. This gives an objective measure of the perceptual quality of the decoded sequences. Distortion is computed as a distance between the original and reconstructed pictures.

A video frame is composed by M x N pixels, where M is the length and N is the height of the frame. Each pixel is presented by one luminance value and a set of pixels by two chrominance values. Because the human eye is more sensible to the change in luminance we focus only on this parameter. The MSE and the PSNR in decibels are computed by the following two equations:

$$MSE = \frac{\sum_{\forall i,j} [f(i,j) - F(i,j)]^2}{M \bullet N}$$
(5.2)

$$PSNR = 10 \bullet \log_{10} \left(\frac{255^2}{MSE} \right) = 20 \bullet \log_{10} \left(\frac{255}{\sqrt{MSE}} \right)$$
(5.3)

Where f(i,j) represents the original source frame and F(i,j) represents the reconstructed error-prone frame containing M by N pixels. As mentioned before, wireless links have smaller bandwidth than wired links. Therefore we concentrate on the transmission of videos in the QCIF format (M=176,N=144). In our experiments, we focus only on the luminance parameter for the calculation of the PSNR.



Figure 5.17. PSNR Comparison for the "Suzie" Sequence

To estimate the quality through the PSNR value, the error-prone stream is compared with the original stream frame by frame. In the case of missing frames we have to insert dummy frames that are copies of the last successful frame to ensure that the YUV streams will not lose synchronization. Figure 5.17 shows the results of PSNR performance comparison for the "Suzie" sequence. With the increase of packet error rate, the average PSNR of the video sequence decreases, which is in agreement with its subjective measure. That is, the image quality degrades with the increase of packet error rate. Furthermore, Our proposed ADSR-ARQ scheme improves PSNR by 3 dB on average compared to the case without ARQ. The tests demonstrate that video image quality degradation increases with the increase of packet error/loss and ADSR-ARQ error control scheme can greatly improve the overall video quality by reducing the packet losses.

5.2.5 Summary of Experiments

In the first test it was verified the two-state Markov error model developed in our wireless emulator. From those burst error patterns, it shows that this two-state Markov chain

error model can model the burst error behavior as we desired. Next, we investigated the ROHC header compression performance. The end-to-end delay with ROHC is significantly reduced, and its corresponding throughput is greatly improved. The results demonstrate that the ROHC header compression scheme provides high bandwidth efficiency and improves the performance of video streaming over bandwidth limited wireless channels. Then, in another test it was observed that a minimum bandwidth is necessary to guarantee an acceptable video quality for video streaming over wireless channels. It also shows MPEG-4 encoded video stream can be transmitted through the low bandwidth of IS95 cellular systems.

In the last test we evaluated the performance of our proposed ADSR-ARQ error control scheme. Compared to the conventional SR-ARQ, ADSR-ARQ reduced average end-to-end packet delay by 60% and increased average throughput by 35% under error condition of 10% PER and average burst length of 7. The delay jitter with ADSR-ARQ is reduced dramatically compared with SR-ARQ. In addition, ADSR-ARQ outperformed the conventional SR-ARQ under different channel and error conditions. It was observed that the ADSR-ARQ greatly improves the overall video quality by trade-off the packet loss and delay. In other words, ADSR-ARQ is more suited for real time video transmission over wireless channels than the conventional SR-ARQ.

Chapter 6: Conclusions and Future Work

Video streaming applications are delay sensitive, which usually have delay constrained QoS requirements. Streaming video over error-prone and bandwidth limited wireless channels faces a lot of challenges. From the network's point of view, conventional retransmission based error control schemes yield a long delay, which may not be suitable for real-time applications.

In this thesis, a LAN-based real-time emulated wireless MPEG-4 video streaming testbed was built to allow us to investigate the performance of video streaming applications over wireless networks. The wireless emulator was implemented by NS2 to simulate the characteristics of CDMA cellular systems, such as packet loss and delay. A two-state Markov channel model was developed to model wireless error behaviors in the wireless emulator. To improve the bandwidth efficiency, the Robust Header Compression (ROHC) scheme was introduced into the wireless emulator as well. A pseudo ROHC function was inserted into the Radio Link Layer of the emulator. Experiment results shows that there are 11.6% throughput gain and 17.5% overall packet delay reduction by using ROHC. The ROHC header compression is especially useful for low-bandwidth networks, such as IS-95 systems.

To optimize the performance of MPEG-4 video streaming over 3G wireless networks without modifying the existing applications, this thesis proposed a novel Adaptive Delayconstrained Selective Repeat ARQ (ADSR-ARQ) error control scheme in the radio link layer, which is able to dynamically switch ARQ "on" or "off" and limits the retransmission attempts based on the transmission buffer length. Combining an optimal design of cumulative NACK with a packet scheduling, this scheme can achieve an overall acceptable video quality. Our work shows that ADSR-ARQ can reduce end-to-end delay and jitter, as well as improve the

Chapter 6: Conclusions

channel throughput. As we expected, the ADSR-ARQ overcomes the drawbacks of conventional SR-ARQ and provides significant improvements on performance of MPEG-4 video streaming over CDMA wireless networks.

One of the contributions of this thesis was in the design and implementation of the core of the wireless emulator, including channel model, header compression scheme, queue management and packet scheduling, as well as building a LAN-based real-time wireless multimedia testbed. Another contribution was to perform measurements to study and evaluate the performance of MPEG-4 video streaming over CDMA wireless networks. In particular, a new ADSR-ARQ error control scheme was proposed to optimize wireless video transmission performance, which provides wireless service providers an insight of performance enhancement for error prone wireless environments.

Due to some limitations of the MPEG-4 streaming server and IM1 reference software, further performance evaluation and optimization of wireless video stream transmission were not possible to do. Since the wireless emulator we implemented is content independent, and our proposed ADSR-ARQ scheme also can be used for other real-time traffics, this wireless emulator can be used to study and investigate new algorithms for future streaming media applications over a CDMA wireless link with variation in system parameters and channel conditions.

Bibliography

- [1] J. Hunter, V. Witana and M. Antoniades, "A Review of Video Streaming over the Internet," *DSTC Technical Report TR97-10*, August 1997 <u>http://archive.dstc.edu.au/RDU/staff/jane-hunter/video-streaming.html</u>
- [2] E. Berruto, M. Gudmonson, R. Menolascino, W. Mohr, and M. Pizarroso, "Research Activities on UMTS Radio Interface, Network Architectures, and Planning," *IEEE Commun. Mag.*, vol. 36, no. 2, Feb. 1998, pp. 82-95.
- [3] D. Grillo, ed., "Special Section on Third-Generation Mobile Systems in Europe," *IEEE Personal Commun. Mag.*, vol. 5, no.2, April 1998, pp. 5-38.
- [4] B. Girod and N. Farber, Compressed Video Over Networks, Chapter 12: Wireless Video, Marcel Dekker Inc, Nov. 1999, pp. 2-6.
- [5] H. Chaskar, "Requirements of a QoS Solution for Mobile IP," IETF draft-ietf-mobileipqos-requirements-00.txt, Jun 2001
- [6] L. Westberg and M. Lindqvist, "Realtime Traffic over Cellular Access Networks," *IETF draft-westberg-realtime-cellular-04.txt*, June 2001
- [7] W. Richard Stevens, *TCP/IP Illustrated, Volume 1, The Protocols*, Addison-Wesley, 1994, pp. 3-4.
- [8] P. Bahl and B. Girod, eds., "Special Section on Wireless Video," *IEEE commun. Mag.*, vol. 36, no. 6, June 1998, pp. 92-151.
- [9] S. Lin, D.J. Costello, and M.J. Miller, "Automatic repeat error control schemes," *IEEE Commun. Mag.*, vol. 22, Dec. 1984, pp. 5-17.
- [10] A. Heron and N. MacDonald, "Video transmission over a radio link using H.261 and DECT," *in Inst. Elect. Eng. Conf. Publications*, no. 354, 1992, pp. 621-624.
- [11] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, "Low Bit-Rate Video Transmission over Fading Channels for Wireless Microcellular Systems," *IEEE Trans.* On Circuits and Systems for Video Technol., vol. 6, no. 1, Feb. 1996, pp. 1-11.
- [12] D. Wu, Y.T. Hou, W. Zhou, H. Lee, T. Chiang, Y Zhang, and H.J. Chao, "On End-to-End Architecture for Transporting MPEG-4 Video over the Internet," *IEEE Transactions On Circuits And Systems For Video Technology*, Vol. 10, No.6, Sep 2000
- [13] D. Wu, Y.T. Hou, Y. Zhang, "Transporting Real-time Video over the Internet: Challenges and Approaches," *Proceedings of the IEEE*, Vol. 88, No. 12, Dec 2000
- [14] S. Wang, H. Zheng and J.A. Copeland, "An Error Control Design for Multimedia

Wireless Network," *The IEEE Annual Vehicular Technology Conference* (VTC2000-Spring), Tokyo, Japan, May 2000.

- [15] C.H. Wang, R.I. Chang, J.M. Ho and S.C. Hsu, "Rate-sensitive ARQ for real-time video streaming," *IEEE Global Communications Conference*, December, 2003.
- [16] S. Aramvith, C.W. Lin, S. Roy and M.T. Sun, "Wireless Video Transport Using Conditional Retransmission and Low-Delay Interleaving," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.12, no.6, June 2002, pp.558-565.
- [17] A. Aguiar, C. Hoene, J. Klaue, H. Karl, H. Miesmer, and A. Wolisz, "Channel-aware Schedulers for VoIP and MPEG4 based on Channel Prediction," In Proc. 8th Intl. Workshop on Mobile Multimedia Communications (MoMuC'03), October 2003.
- [18] J. Chen and V.C.M. Leung, "Applying Active Queue Management to Link Layer Buffers for Real-time Traffic over Third Generation Wireless Networks," *in Proc. IEEE WCNC'03*, New Orleans, LA, Mar. 2003.
- [19] "Streaming Methods: Web Server vs. Streaming Media Server," Microsoft Corporation, October 2001. <u>http://www.microsoft.com/windows/windowsmedia/compare/webservvstreamserv.asp</u>
- [20] "RTP: A Transport Protocol for Real-Time Applications," Network Working Group RFC 3550, July 2003. <u>http://rfc.sunsite.dk/rfc/rfc3550.html</u>
- [21] J. Postel, "Transmission Control Protocol," RFC 0793, IETF, September 1981. http://www.faqs.org/rfcs/rfc793.html
- [22] J. Postel, "User Datagram Protocol", RFC 0789, IETF, August 1980. http://www.faqs.org/rfcs/rfc768.html
- [23] D. Hoffman, G. Fernando, V. Goyal and M. Civanlar, "RTP payload format for MPEG1/MPEG2 video," RFC 2250, IETF, January 1998. <u>http://www.faqs.org/rfcs/rfc2250.html</u>
- [24] B. Flinchbaugh, M. Zhou and R. Talluri, "How MPEG-4 Trade-offs Affect Design", *EE Times*, November 12, 2001.
- [25] "Coding of Audio-Visual Objects Part 6: Delivery Multimedia Integration Framework (DMIF)," ISO/IEC 14496-6, International Standard, ISO/IEC/SC29/WG11 N2501, March 2000.
- [26] Y. Pourmohammadi, K. Asrar Haghighi, A. Kaheel, H.M. Alnuweiri and S.T. Vuong, "On the Design of a QoS-aware MPEG-4 Multimedia Server", *International Symposium* on telecommunications (IST2001).
- [27] Y. Pourmohammadi, K. Asrar Haghighi and H.M. Alnuweiri, "Internet delivery of

MPEG-4 object-based multimedia," *IEEE Multimedia Magazine*, vol 10, no. 3, July-Sept. 2003, pp. 68 - 78.

- [28] Y. Pourmohammadi, K. Asrar Haghighi, A. Mohamed and H.M.Alnuweiri, "Streaming MPEG-4 over IP and Broadcast Networks: DMIF based Architectures," *PacketVideo* 2001.
- [29] T.S. Rappaport, B.D. Woerner and J.H. Reed, *Wireless Personal Communications*, Kluwer Academic Publishers, 1996.
- [30] R.L. Pickholtz, D.L. Schilling and L.B. Milstein, "Theory of Spread Spectrum Communications - A Tutorial," *IEEE Trans. Commun.*, vol. 30, no. 5, May 1982, pp. 855-884.
- [31] TIA/EIA/IS-95 Rev A, "Mobile Station Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular Systems," Washington: Telecommunication Industry Association, 1995.
- [32] TIA/EIA/cdma2000, "Mobile Station Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular Systems," Washington: Telecommunication Industry Association, 1999.
- [33] P.A. Chou, A. Mohr, A. Wang and S. Mehrota, "Error control for receiver-driven layered multicast of audio and video," *IEEE Trans. on Multimedia*, vol. 3, Mar. 2001, pp. 108–122.
- [34] E.N. Gilbert, "Capacity of a burst Noise Channel," *Bell System Tech. Journal*, Vol. 39, Sep. 1960, pp. 1253-1266
- [35] C. Jiao, L. Schwiebert and B. Xu, "On Modeling the Packet Error Statistics in Bursty Channels," 27th Annual IEEE conference (LCN'02), Nov. 2002.
- [36] L.N. Kanal and A.R.K. Sastry, "Models for channels with memory and their applications to error control," *Proc. Of the IEEE*, vol. 66, no. 7, July 1978, pp. 724-744.
- [37] C.C. Tan and N.C. Beaulieu, "On First-Order Markov Modeling for the Rayleigh Fading Channel," *IEEE Transactions on Communications*, Vol. 48, No. 12, December 2000, pp. 2032-2040.
- [38] H.S. Wang and P. Chang, "On Verifying the First-Order Markovian Assumption for a Rayleigh Fading Channel Model," *IEEE Transactions on Vehicular Technology*, Vol. 45, No. 2, May 1996, pp. 353-357.
- [39] UCB/LBNL/VINT, "Network Simulator (version 2)," August 2002. http://www.isi.edu/nsnam/ns/
- [40] Z. Yin and V.C.M. Leung, "A Proxy Architecture to Enhance the Performance of WAP 2.0 by Data Compression," *in Proc. IEEE WCNC'03*, New Orleans, LA, Mar. 2003.

- [41] F. Fitzek, S. Hendrata, P. Seeling and M. Reisslein, "Video Quality Evaluation for Wireless Transmission with Robust Header Compression," *Technical Report acticom-03-003*, July, 2003
- [42] H. Naser, A. Leon-Gracia, "Performance evaluation of MPEG2 video using guaranteed service over IP-ATM networks," *Multimedia Computing and Systems Conference*, Austin, TX, USA, Jun. 1998.
- [43] A. Heron and N. MacDonald, "Video transmission over a radio link using H.261 and DECT," in *Inst. Elect. Eng. Conf. Publications*, no. 354, 1992, pp. 621-624.
- [44] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, "Low Bit-Rate Video Transmission over Fading Channels for Wireless Microcellular Systems," *IEEE Trans.* On Circuits and Systems for Video Technol., vol. 6, no. 1, Feb. 1996, pp. 1011.
- [45] S. Lin, D.J. Costello and M.J. Miller, "Automatic repeat error control schemes," *IEEE Commun. Mag.*, vol. 22, Dec. 1984, pp. 5-17.
- [46] F. Hartanto and H.R. Sirisena, "Hybrid error control mechanism for video transmission in the wireless IP networks," *IEEE LANMAN'99 Post-Conference Proc.*, 2001
- [47] D.G. Sachs, I. Kozintsev, M. Yeung, D.L. Jones, "Hybrid ARQ for Robust Video Streaming over Wireless LANs," *International Conference* (ITCC2001), Las Vegas, NV, April 2001.
- [48] M. Zorzi, "Performance of FEC and ARQ Error Control in Bursty Channels under Delay Constraints," *in Proc. IEEE VTC* '98, 1998, pp. 1390-1394.
- [49] K. Norbert, "Radio Link Protocol (RLP) for Circuit Switched Bearer and Teleservices," UMTS, 3GPP TS 24.022 version 4.0.0 Release 4.
- [50] F.H.P. Fitzek, A. Köpsel, A. Wolisz, M. Reisslein and M. A. Krishnam, "Providing Application-Level QoS in 3G/4G Wireless Systems: A Comprehensive Framework Based on Multi-Rate CDMA," in IEEE International Conference on Third Generation Wireless Communications, Jun. 2001, pp. 344–349.
- [51] C. Bormann, et.al., "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed," *Technical report, RFC 3095*, July 2001.
- [52] R. Jain, Extending the ns2 simulator for GPRS support, master's thesis, Dept. EE, Indian Inst. of Tech. Mumbai, India. <u>http://www.it.iitb.ac.in/research/groups/mobile_computing/2001/abstracts.php3#richa</u>
- [53] "An introduction to Reed-Solomon codes: principles, architecture and implementation," http://www.4i2i.com/reed_solomon_codes.htm
- [54] M. Zorzi and R. R. Rao, "Lateness probability of a retransmission scheme for error control on a two-state Markov channel", *IEEE Trans. Commun.*, vol. 47, Oct. 1999. pp. 1537-1548.

Bibliography

[55] "Extensions to the *ns* Network Simulator," <u>http://www.icsi.berkeley.edu/~widmer/mnav/ns-extension/</u>