A Novel Approach for Implementing Worldwide Interoperability for Microwave Access for Video Surveillance

Suherman

Faculty of Technology, De Montfort University

A thesis submitted in partial fulfilment of the

requirements of De Montfort University for the degree of

Doctor of Philosophy

July 2013

Abstract

Video surveillance applications have experienced an increase in demand over the last decade. Surveillance systems can easily be found in places such as commercial offices, banks and traffic intersections, parks and recreational areas. Surveillance applications have the potential to be implemented on a WiMAX (Worldwide Interoperability for Microwave Access) network. Moreover, WiMAX devices have been used widely in the market and WiMAX-based video surveillance products have also been available. As a radio technology, WiMAX is a wireless broadband system that offers greater capacity than WiFi networks and wider coverage than cellular networks.

The acceptance of WiMAX in the market, the availability of WiMAX products and its technology excellence, contribute to the possibility of implementing it for surveillance application. However, since WiMAX is designed to accommodate various applications with different quality of service (QoS) requirements, dedicated surveillance network implementation of WiMAX may not achieve optimum performance, as all Subscriber Stations (SSs) generate the same QoS requirements.

In the medium access (MAC) layer, this thesis proposes a bandwidth allocation scheme that considers the QoS uniformity of the traffic sources. The proposed bandwidth allocation scheme comprises a simplified bandwidth allocation architecture, a packet-aware bandwidth request mechanism and packet-aware scheduling algorithms. The simplified architecture maximizes resources in the Base Station (BS), deactivates unnecessary services and minimizes the processing delay. The proposed bandwidth request mechanism reduces bandwidth grant and transmission delays. The proposed scheduling algorithms prioritize bandwidth granting access to a request that contains important packet(s). The proposed methods in the MAC layer are designed to be applied to existing devices in the market, without the necessity to change hardware.

The transport protocol should be able to deliver video with sufficient quality while maintaining low delay connectivity. The proposed transport layer protocol is therefore designed to improve the existing user datagram protocol (UDP) performance by

retransmitting packet loss selectively to increase the received video quality, and utilizing MAC support to achieve low delay connectivity.

In order to overcome the limitations of the lower layers, this thesis employs a rateless code instead of transport layer redundancy in the application layer. Moreover, this thesis proposes post-decoding error concealment techniques as the last means to overcome packet loss.

To evaluate the performances of the proposed methods, simulations are carried out using NS-2 simulator on Linux platform. The proposed methods are compared to existing works to measure their effectiveness. To facilitate the implementation of the transport layer protocols in practical scenarios, UDP packet modification is applied for each transport layer protocol.

Declaration

I declare that the material presented in this thesis consists of original work undertaken solely by myself. Information derived from the published and unpublished work of others has been properly referenced. The material has not been submitted in substantially the same form for the award of a higher degree elsewhere.

Suherman

May 2013

List of Publications:

This thesis is based on the following publications:

Journals:

1. Suherman S., Al-Akaidi M., *Increasing Uplink Broadband Video Streaming Performance in WiMAX Networks*, International Journal of Internet Protocol, vol. 7 no. 3, pp. 176-185, 2012.
2. Suherman S., Al-Akaidi M., *Adjusting WiMAX for a dedicated surveillance network*, International Journal of Electrical and Computer Engineering, vol 3, no 4, pp. 492-503, August 2013.
3. Suherman S., Al-Akaidi M., *An efficient negative acknowledgement-based transport protocols in 802.11 media streaming*, International Journal of Ad Hoc and Ubiquitous Computing (accepted on 22nd April 2013).
4. Suherman S., Al-Akaidi M., *A transport layer protocol for WiMAX mobile terminal based video streaming*, International Journal of Handheld Computing Research (under revision).

Conferences:

1. Suherman S., Al-Akaidi M., Hamzaoui R., *Transport and MAC Cross Layer Protocol For WiMAX Based Dedicated Video Surveillance Network*, in the proceeding of the 13th Middle Eastern Simulation and Modeling Conference, Muscat, Oman, Dec 10–12, 2012, pp 52-59.
2. Suherman S., Al-Akaidi M., Hamzaoui R., *Inter-frame Retransmission for Video Surveillance over WiMAX*, in the proceeding of the 12th Middle Eastern Simulation and Modeling Conference, Amman, Jordan, Nov 2011, pp 42-47.
3. Suherman S., Al-Akaidi M., *Improving Real Time Video Surveillance Performance Using Inter-frame Retransmission*, in the International Conference on Imaging for Crime Detection and Prevention, London, Nov 2011.

4. Suherman S., Al-Akaidi M., *Post-decoding Concealment Techniques for Video Transmission*, in the Global Conference of Communication Science and Information Engineering, London, July 2011.

Acknowledgements

I would like to thank, first of all, Allah, the God almighty, which guide my life with Rahmah and Rizqi. My sincerest gratitude goes to my supervisor, Professor Marwan Al-Akaidi for his kindness that helped me to choose this topic and also for his guidance throughout my PhD study. His thoughtful guidance and insightful suggestions played a key role in the completion of this thesis. I also greatly appreciate his care and concern not only for my intellectual and personal growth but also for the well-being of my family and myself.

I would like to thank Professor Raouf Hamzaoui for his constructive criticism and guidance throughout the research work. I want also to thank members of the Multimedia Communication and Signal Processing group within the Faculty of Technology especially Dr. Waleed Al-Azzawi, Dr. Ahmad Shakeel, and Dr. Muneeb Dawood for valuable discussions.

I thank my sponsor, the Indonesian Directorate General of Higher Education (DGHE/DIKTI) and also my employer, the University of Sumatera Utara for providing me with financial support for the duration of my PhD research.

I would like to thank my parents for their love and prayers throughout my life. I would also like to thank my beloved wife, Sariana and my sweet bones, Yasmin Fajri, Anis Lee and Aisha Ahmad who had always been by my side giving their love, encouragement, and support through my tough times. This thesis is dedicated to them.

Table of Contents

Abstract ii
Declaration iv
List of Publications: v
Acknowledgements vii
Table of Contents viii
List of Tables xii
List of Figures xiii
List of Acronyms xv
Chapter 1 Introduction 1
1.1 Motivation 1
1.2 Problem formulation 2
1.3 Multi-layer approach 2
1.4 Aims and objectives 3
1.5 Scope of the thesis 3
1.6 Contributions of the thesis 4
1.7 Outline of the thesis 5
Chapter 2 Background 7
2.1 Surveillance technologies 7
2.2 Communication protocol 8
2.3 MAC layer: WiMAX systems 9
2.3.1 WiMAX architecture 9
a. Connection establishment 11
b. MAC frames 12
c. WiMAX services 13
d. Bandwidth request and bandwidth allocation scheme 14
2.3.2 Research on WiMAX 14
2.4 Transport layer protocols 16
2.4.1 Unreliable transport layer protocol 16
2.4.2 The need to improve the unreliable protocol 17
2.5 Cross-layer techniques 17
2.6 Application layer techniques 18

2.7 WiMAX simulation 19
2.7.1 NS-2 simulator 20
2.7.2 NIST WiMAX Module 20
2.7.3 Channel model 22
2.7.4 Error model 22
2.7.5 Evaluation framework 24
2.7.6 Evaluation metrics 25
2.7.7 Simulation scenario 26
2.8 Summary 29
Chapter 3 Related works 30
3.1 WiMAX for surveillance network 30
3.2 WiMAX bandwidth request mechanisms 31
3.3 WiMAX scheduling algorithms 33
3.4 Unreliable transport layer protocol 35
3.5 WiMAX Cross-layer techniques 38
3.6 Error concealment techniques 40
3.7 Summary 41
Chapter 4 MAC layer: Bandwidth allocation 42
4.1 Introduction 42
4.2 Proposed methods 43
4.2.1 Flat service class architecture 43
4.2.2 Packet-aware bandwidth request 46
4.2.3 Packet-aware non-sorting schedulers 50
4.3 Performance evaluation 52
4.4 Result and analysis 53
4.4.1 Performance of the flat BE service architecture 53
4.4.2 Analysis of bandwidth request mechanisms 54
4.4.3 Analysis of Scheduling Algorithms 57
4.5 Summary 59
Chapter 5 Transport layer: UDP improvement 61
5.1 Introduction 61
5.2 Inter-frame retransmission protocol 61
5.2.1 Negative acknowledgement 62
5.2.2 Retransmission scheduling 62
5.2.3 Prioritized packets 65

5.2.4 Congestion delay 65
5.2.5 Unicast CD for frequent retransmission 66
5.2.6 Performance evaluation 67
5.2.7 The protocol performance 67
5.2.8 Performance comparisons 71
5.2.9 Unicast CD performance 74
5.2.10 Inter-frame retransmission in other network (802.11) 75
5.3 Transport and MAC cross-layer protocol 76
5.3.1 Early bandwidth request 78
5.3.2 The impact of MAC cross-layer 79
5.3.3 Transport protocol comparisons 82
5.3.4 Protocol performance over various schedulers 84
5.3.5 Protocol performance over previously proposed schedulers 87
5.3.6 Cross-layer impact to other protocol 88
5.4 Summary 92
Chapter 6 Application layer: redundancy and error concealment 93
6.1 Introduction 93
6.2 Redundancy techniques 93
6.2.1 Transport-layer redundancy 94
6.2.2 Application layer redundancy 94
6.2.3 Redundant packets transmission 95
6.2.4 Performance evaluation 95
6.2.5 Performance degradation 95
6.2.6 Redundancy performance 97
6.3 Post-decoding error concealment techniques 100
6.3.1 Inline FC method 100
6.3.2 Closed related FC method 101
6.3.3 Referenced partial FC method 102
6.3.4 Combined partial FC method 103
6.3.5 Pattern partial FC method 104
6.3.6 Evaluation method and results 105
6.4 Summary 107
Chapter 7 Mathematical model and analysis 108
7.1 Introduction 108
7.2 Application layer model 108

7.3 Transport layer model 110
7.4 WiMAX network model.. 112
7.5 Packet loss 113
7.6 Packet delay.. 113
7.7 Results of the analysis 114
7.8 Summary.. 117
Chapter 8 Experimental performance evaluation 119
8.1 Introduction.. 119
8.2 Experimental network 120
8.3 Transport layer protocol 121
8.3.1 Video streamer 121
8.3.2 Experiment results.. 122
8.4 Transport layer protocol performance in 802.11 networks 123
8.5 Post-decoding error concealment techniques 124
8.5.1 Streamer design.. 124
8.5.2 Experiment results.. 126
8.6 Summary.. 128
Chapter 9 Conclusion.. 130
9.1 Summary.. 130
9.2 Future work.. 132
9.2.1 MAC layer 132
9.2.2 Transport layer protocol.. 133
References 134
Appendix 1The NS-2 implementation xv
1. The modified files xv
2. Transport layer protocol implementation.. xvi
2.1. Inter-frame retransmission. xvi
2.2. Congestion delay xxi
2.3. Transport layer redundancy xxiii
3. MAC layer implementation..xxiii
3.1. Transport-MAC cross-layer protocol xxiii
3.2. Bandwidth request mechanism xxv
3.3. Scheduling algorithms xxix

List of Tables

Table 2.1: Simulated traffic parameters 29
Table 4.1: Performance of scheduling algorithms using contention request 57
Table 5.1: Packet loss comparison 75
Table 5.2: Bandwidth comparison for GOP 30 80
Table 6.1: Simulation parameters 105
Table 6.2: The average PSNR over erroneous frames 106
Table 8.1: The observed video in SS2 125
Table 8.2: PSNR Performance of post-decoding error concealment 128

List of Figures

Figure 1.1: Surveillance network components ... 2
Figure 2.1: WiMAX frame structure ... 9
Figure 2.2: WiMAX MAC architecture [17] ... 10
Figure 2.3: Connection establishment ... 12
Figure 2.4: UDP header, from [35] ... 16
Figure 2.5: TCP/IP data encapsulation ... 17
Figure 2.6: The NIST WiMAX module[40] ... 21
Figure 2.7: The 2-state Markov model ... 23
Figure 2.8: The Evalvid framework ... 24
Figure 3.1: Existing NACK scheduling ... 38
Figure 4.1: Bandwidth allocation architectures ... 45
Figure 4.2: Performance of TBEB ... 47
Figure 4.3: Piggybacked and wasted bandwidth ... 49
Figure 4.4: RR based packet-aware non-sorting scheduler ... 51
Figure 4.5: FIFO-based scheduler algorithm ... 52
Figure 4.6: Performance of BE vs rtPS ... 54
Figure 4.7: Performance of the predicted next frame piggybacking ... 55
Figure 4.8: Performance of bandwidth request mechanisms ... 56
Figure 5.1: Negative acknowledgement ... 62
Figure 5.2: Existing NACK scheduling ... 62
Figure 5.3: NACK scheduling ... 64
Figure 5.4: Inter-frame retransmission with CD ... 66
Figure 5.5: Unicast CD request ... 67
Figure 5.6: Comparison of QR and IR performance ... 68
Figure 5.7: Comparison of IR and prioritized IR performance ... 69
Figure 5.8: Congestion delay performances ... 70
Figure 5.9: Congestion delay impact ... 70
Figure 5.10**:** Performance comparisons of transport protocols ... 71
Figure 5.11: Performance comparisons ... 73
Figure 5.12: IR with unicast CD performance ... 74
Figure 5.13: TCP performance ... 75
Figure 5.14: Packet loss and delay ... 76
Figure 5.15: MAC layer implementation ... 78
Figure 5.16: Subscriber station based MAC cross-layer ... 79
Figure 5.17: The requested and the allocated bandwidth for GOP 30 ... 81
Figure 5.18: Performance comparison between IR and TMC ... 82
Figure 5.19: Performance comparison between TMC and other protocols ... 83
Figure 5.20: Comparison of the allocated bandwidth and network utility ... 84

Figure 5.21: TMC performances over various schedulers 86
Figure 5.22: TMC performances for packet-aware schedulers 87
Figure 5.23: Cross-layer impact on bandwidth and buffer 88
Figure 5.24: Cross-layer impact on packet loss.. 89
Figure 5.25: Cross-layer impact on delay and PSNR 90
Figure 5.26: Protocol performance comparisons... 91
Figure 6.1: Packet redundancies 94
Figure 6.2: Video surveillance performances in error-prone environment.................. 96
Figure 6.3: Delay in error-prone environment 97
Figure 6.4: Packet loss in error-prone environment 98
Figure 6.5: PSNR in error-prone environment.. 98
Figure 6.6: Rateless code performance for various PE 99
Figure 6.7: Frame copy error concealment 100
Figure 6.8: Post-decoding FC performance 105
Figure 7.1: Application layer model.. 109
Figure 7.2: Transport layer protocol model 111
Figure 7.3: Request slots and their allocation 112
Figure 7.4: Protocol performances for various numbers of SSs................................ 115
Figure 7.5: Transport layer protocol performances 116
Figure 8.1: Transport layer protocol performances 121
Figure 8.2: Packet format of the transport protocols 121
Figure 8.3: Protocol performance comparisons... 122
Figure 8.4. Diagram of the experimental 802.11 network... 123
Figure 8.5: Packet loss and delay in an 802.11 network.. 124
Figure 8.6: Header format 125
Figure 8.7: The packet loss pattern; ‘1’ = accepted, ‘0’ = lost.................................. 126
Figure 8.8: Rate variation... 127
Figure 1: The sender and receiver methods of the proposed IR................................. xxi

List of Acronyms

3GPP	3rd Generation Partnership Project
ACK	Acknowledgement
aPS	adaptive Polling Service
AVC	Advanced Video Coding
AVI	Audio Video Interleave
BC	Block Copy
BE	Best Effort
BS	Base Station
BVS	Broadband Video Streaming
BW	Bandwidth
CAC	Call Admission Control
CBR	Constant Bit Rate
CCTV	Closed-Circuit TeleVision
CD	Congestion Delay
CDMA	Code Division Multiple Access
CID	Connection Identification
CIF	Common Intermediate Format
COST	COopération européenne dans le domaine de la recherche Scientifique et Technique
CRC	Cyclic Redundancy Check
CW	Contention Window
DAA	Detection and Avoidance Algorithm
dB	deciBel
DCCP	Datagram Congestion Control Protocol
DL	Downlink
DRR	Deficit Round Robin
EDF	Earliest Deadline First
EIED	Exponential Increase Exponential Decrease
ertPS	enhanced real-time Polling Service

FC	Frame Copy
FDD	Frequency Division Duplex
FIFO	First In First Out
GOP	Group of Pictures
GPC	Grant Per Connection
GPSS	Grant Per Subscriber Station
HMVE	Hybrid Motion Vector Estimation
HSPA	High-Speed Packet Access
IFG	Inter-frame Gap
IFQ	Interface Queue
IP	Internet Protocol
IR	Inter-frame Retransmission
ISM	Industrial, Scientific and Medical
ISO	International Organization for Standardization
LTE	Long-term Evolution
MAC	Medium Access Layer
Mbps	Mega bit per second
MN	Mobile Node
MOS	Minimum Opinion Score
MoSES	MObile Streaming for vidEo Surveillance
MPEG	Moving Picture Experts Group
MTU	Minimum Transmission Unit
MV	Motion Vector
NACK	Negative ACKnowledgement
NCTU	National Chiao Tung University
NDSL	Networks and Distributed System Laboratory
NIST	National Institute of Standards and Technology
nrtPS	non real-time Polling Service
NTP	Network Time Protocol
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OFFC	Object-based Full Frame Concealment

OS	Operating System
OSI	Open Systems Interconnection
OTcl	Object-oriented Tool command language
PDU	Packet Data Unit
Pe	Probability of error
PHY	Physical
PMP	Point-to-multipoint
PSNR	Peak Signal-to-Noise Ratio
QCIF	Quarter Common Intermediate Format
QoS	Quality of Service
QP	Quantization Parameter
RBUDP	Reliable Blast User Datagram Protocol
REQ	Request
RTG	Receive/transmit Time Gap
rtPS	real-time Polling Service
RUDP	Reliable User Datagram Protocol
SCTP	Stream Control Transmission Protocol
SDU	Service Data Unit
SIP	Session Initiation Protocol
SS	Subscriber Station
SSA	Spectrum Sensing Algorithm
SUI	Stanford University Interim
TBEB	Truncated Binary Exponential Backoff
TCP	Transmission Control Protocol
TDD	Time Division Duplexing
TDMA	Time Division Multiple Access
TMC	Transport MAC Cross-layer
TTG	Transmit/receive Time Gap
UBB	Utility Based Backoff
UDP	User Datagram Protocol
UGS	Unsolicited Grant Scheme
UL	Uplink

UWB	Ultra Wide Band
VCR	Video Cassette Recorder
WCCP	Wireless Connection Control Protocol
WFQ	Weighted Fair Queuing
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WRR	Weighted Round Robin

Chapter 1
Introduction

1.1 Motivation

Video surveillance is an emerging application for activity and security monitoring. Outdoor surveillance applications can take advantage of a WiMAX network to provide installation flexibility and mobility. A WiMAX-based surveillance system can be implemented as a dedicated network serving only surveillance nodes to ensure high reliability. WiMAX technology offers high bandwidth connectivity and user mobility. As a broadband technology, WiMAX is able to provide a cell bit rate greater than 100 Mbps, covering an area up to 50 km [1]. Such technology makes it possible to deliver high bit rate video application. Moreover, the mobility feature of WiMAX enables video surveillance to be attached to moving objects, such as public transportation or security patrols. The increasing number of surveillance application requires a broadband infrastructure.

A WiMAX network is designed for multi-services, ranging from data to real-time applications, and low priority to higher priority. In current WiMAX architecture, a real-time multimedia application is served by rtPS service, which requires QoS negotiation and the enforcement of traffic parameters [2]. When all subscriber nodes are intended for surveillance cameras that generate video traffic, the result is a high network load, over-utilizing rtPS service and under-utilizing other services; consequently, there is a waste of network resources. Due to the nature of similar traffic types in surveillance application, there is an adjustment requirement in current WiMAX network to be used as an infrastructure for the surveillance network.

Furthermore, existing bandwidth allocations in WiMAX are designed in accordance with a multi-service platform; therefore, the performance may not be optimum for similar traffic-type services. An alternative solution is required for the surveillance application.

1.2 Problem formulation

The use of WiMAX in a dedicated video surveillance network is realized by attaching surveillance cameras to subscriber stations (SSs). Video packets are generated by the SSs, transmitted over the WiMAX link and forwarded by the base station (BS) to the monitoring station through other types of network(s).

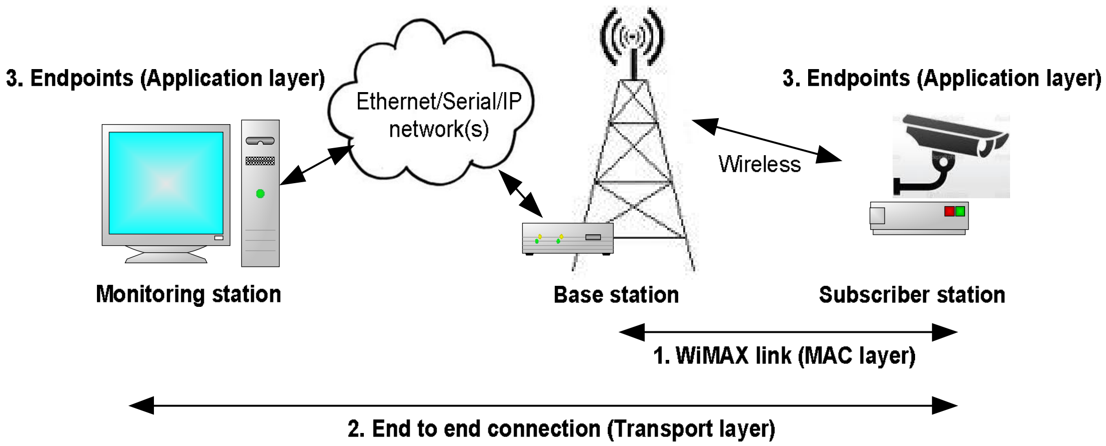


Figure 1.1: Surveillance network components

There are three important entities involved in the WiMAX-based surveillance network depicted in Figure 1.1: WiMAX link; end-to-end connection; and endpoints. The three entities lie on different layers: MAC layer; transport layer; and application layer. Therefore, a multi-layer approach is required in order to enhance the surveillance performance.

1.3 Multi-layer approach

Layered architecture represents a communication network in different layers. For instance, physical hardware is defined in physical layer. The goal is to simplify network understanding and interoperability between different communication devices from different vendors.

This thesis uses a multi-layer approach to improve WiMAX performance for dedicated video surveillance networks. The multi-layer approach comprises three layers, including a MAC layer to improve WiMAX link, a transport layer to improve the end-to-end performance and the application layer as the last solution to overcome lower-layer limitations.

Although the IEEE 802.16 standard defines the signalling mechanisms for information exchange between BS and SS, including connection set-up, bandwidth request and MAP messages, the specification does not define a scheduling algorithm that allocates resources at the base station. Furthermore, the WiMAX bandwidth request-grant scheme is implementation-dependent. The MAC layer solution goal is to optimize the WiMAX link by using a particular bandwidth request mechanism and scheduling algorithm.

The transport layer protocol enhances the existing unreliable protocol; whereby the goal is to reduce packet loss without injecting excessive delay. The goal of the application layer solution is to overcome the limitations of the lower layers.

1.4 Aims and objectives

The proposed techniques should meet the following aims and objectives:

- The MAC layer methods should reduce delay and packet loss in the WiMAX link data transfer.
- The MAC layer methods should be applicable to WiMAX devices available in the market without necessary changes to the hardware.
- The transport layer protocol should reduce delay and packet loss in end-to-end connections.
- The transport layer protocol should improve the UDP performance and extend its packets.
- The application layer methods should enhance the video quality without requiring the lower layer to support additional work.

1.5 Scope of the thesis

This thesis proposes the MAC, transport and application layer techniques to improve WiMAX performance for dedicated video surveillance networks.

In the WiMAX network, service types, bandwidth request mechanisms and scheduling algorithms are tuned in accordance with the surveillance traffic. The idea is to consider video packet priority, to reduce bandwidth request delay, and to enhance scheduling performance.

In transport layer protocol design, protocol also prioritizes the important packets, which have a greater impact on image quality. The designed protocol also takes advantage of the link support through the cross-layer schema.

The application layer solutions are designed to overcome lower-layer limitations. Therefore, application layer retransmission is avoided as it burdens the lower layers. Instead, the application layer should be able to improve the received video quality based on the accepted packets.

1.6 Contributions of the thesis

The main contributions of this thesis are:

- A flat service architecture for similar traffic patterns which allocates similarly resources to all incoming traffics. The service is selected based on its simplicity to reduce base-station load.
- A bandwidth request scheme that facilitates fast bandwidth allocation. The fast request scheme combines contention and piggybacking request mechanisms.
- Scheduling algorithms that allocate bandwidth based on packet priority. The priority selection not only depends on the existing request, but also detects the new request. The flat service architecture, the proposed bandwidth request mechanism and scheduling algorithms are published in [3].
- An efficient retransmission method for video traffic. The method is referred to as ‘inter-frame retransmission’. The proposed retransmission protocol is published in [4, 5].
- A cross-layer technique between the transport and MAC layers to reduce packet-loss retransmission delay. This method uses an early bandwidth request schema to support retransmitted packets and is published in [6].
- Rateless code implementation on the WiMAX-based surveillance network. The powerful code enhances video quality in erroneous environments. The work is included in [6].
- Post-decoding error concealment techniques. Error concealment is applied to the received surveillance video in order to recover lost frames. Some of the proposed methods are assisted by comparison process prior packet transmission to the sender. The proposed techniques are published in [7].

1.7 Outline of the thesis

Chapter 2 provides the necessary background information about WiMAX systems, transport layer protocols and application layer techniques. A brief description of the evaluation method is also provided.

Chapter 3 surveys WiMAX bandwidth request mechanisms, scheduling algorithms, unreliable protocol improvements and error concealment techniques.

Chapter 4 proposes MAC layer improvement on WiMAX for a dedicated surveillance network, comprising flat service architecture, packet-aware bandwidth request mechanism and packet-aware scheduling algorithms. Simulation results show that the proposed techniques yield significant delay and PSNR (Peak Signal-to-Noise Ratio) improvements throughout the existing work for WiMAX-based surveillance networks.

Chapter 5 proposes transport layer protocols that improve the existing unreliable protocol by using effective retransmission to reduce packet loss and explore support from MAC layer through a cross-layer approach. Simulation results show that the inter-frame retransmission reduces packet loss and delays more effectively than the existing retransmission methods. Furthermore, the cross-layer technique achieves significant delay reduction and PSNR improvements over the existing protocols, including the UDP.

Chapter 6 implements redundancy techniques and post-decoding error concealment methods in the application layer. Simulation results show that the application layer redundancy (rateless code) performs better than transport layer redundancy for both delay and PSNR. For error concealment techniques, the proposed post-decoding methods perform better than the basic frame copy method.

In Chapter 7, the transport layer protocols are evaluated through a mathematical model. The model approaches the video traffic using a two-state Markovian model. Each transport layer protocol is presented by the probability values for both I and P frame states. Both application and transport layers are applied to WiMAX model. The analysis shows that the proposed protocols achieve delay and packet loss reductions across the existing protocols.

In Chapter 8, the transport layer protocols and error concealment techniques are accessed experimentally. The java programming language is employed to implement the proposed techniques. Experimental works are conducted over WiMAX and WiFi networks. The experiment results show that the proposed techniques achieve better performance than the existing techniques.

Chapter 9 concludes this thesis.

Chapter 2
Background

2.1 Surveillance technologies

Video surveillance applications have experienced an increase in demand over the last decade. Surveillance systems can easily be found not only in places that are sensitive to safety, such as commercial offices, banks and traffic intersections, but also in other areas, such as in parks and recreational areas.

The surveillance technologies have moved from non-real-time systems, where videos are stored and analyzed when unusual situations arise over a given period of time, to intelligent surveillance systems that employ intelligent analysis for real-time image sequencing without human intervention. The invention of digitally-based camera and recording systems has also shifted surveillance systems from using VCR (Video Cassette Recording) to IP surveillance. Although most research fields on video surveillance are dominated by the application layer process, such as motion detection, classification, tracking and behavioural analysis, surveillance infrastructure research is important to support efficient and robust surveillance networks [8].

Most current Closed-circuit television (CCTV) and IP surveillance systems use coaxial and ethernet cable networks for indoor surveillance. Outdoor surveillance applications rely on wireless LAN and point-to-point radio technologies. Although research on the use of cellular networks for surveillance application exists [9-11], its real implementation is hardly found due to the limitations of the channel bandwidth.

WiMAX is a wireless broadband technology that offers greater capacity than WiFi networks and wider coverage than cellular networks. WiMAX experiences intensive development of the standard from a fixed broadband wireless application [12]; mobile WiMAX [13], up to a standard with 4G capabilities [14]. This makes WiMAX a promising technology for video surveillance infrastructures. Surveillance applications have the potential to be implemented on a WiMAX network, such as multi-surveillance cameras placed on high roof tops in urban areas, high speed point-to-point wireless surveillance, and multi-node rural and mobile surveillance.

2.2 Communication protocol

According to Stalling [15], a communications protocol is a set of rules required to send a Protocol Data Unit (PDU)) from one node to another in a network. A PDU, or data, contains a header and payload. A protocol defines how the header and payload to be written and processed. A protocol is usually written in computer programs that communicate with each other through routine calls. Since communication involves hardware and application software, understanding the protocol process will be complex and confusing. Instead of implementing the protocol program by program, separating the process into specific sub-tasks will ease protocol design. The protocol architecture simplifies protocol modules into a vertical arrangement, while the architecture defines protocol in a layered manner.

The Open Systems Interconnection (OSI) reference model was developed by the International Organization for Standardization (ISO) [16] as a protocol model and framework. The model consists of seven layers:

- Application layer; provides user access to OSI environment
- Presentation layer; defines data presentation (syntax)
- Session layer; manages communication stages from establishment to termination
- Transport layer; provides reliable, transparent transfer of data between endpoints, provides end-to-end error recovery and flow control
- Network layer; ensures the upper layers are independent of data transmission and switching technologies
- Data link layer; provides for the reliable transfer of information across the physical link
- Physical layer; deals with physical medium.

However, most protocols do not implement all the layer separation defined in OSI. The model is then left behind. Conversely, TCP/IP architecture has become more accepted than the OSI model. The TCP/IP suites were mature and well tested, preceding other OSI based protocols. The TCP/IP architecture contains:

- Application layer
- Host-to-host (transport) layer
- Internet (network) layer

- Link layer (MAC and PHY layers).

Since TCP/IP architecture dominates current protocol architectures, the TCP/IP model is used throughout this thesis. The following discussion highlights the MAC layer (WiMAX systems), unreliable transport layer protocol and application layer techniques.

2.3 MAC layer: WiMAX systems

The WiMAX standard was published initially in 2001 [12] and operates on 10 to 66 GHz frequency band; however, this was not suitable for urban areas. The IEEE 802.16a standard [13] enables the physical (PHY) layer to operate in the lower frequency band of 2 to 11 GHz. The 802.16e [14] improves previous versions by enabling the MAC and the PHY layers to support mobility features.

2.3.1 WiMAX architecture

The basic WiMAX network consists of one BS and one or more SSs. There are two modes of operation; point-to-multipoint (PMP) and mesh mode. In PMP mode, all SSs communicate through the BS, while in mesh mode; the SS can communicate directly with each other. This thesis focuses on the PMP network.

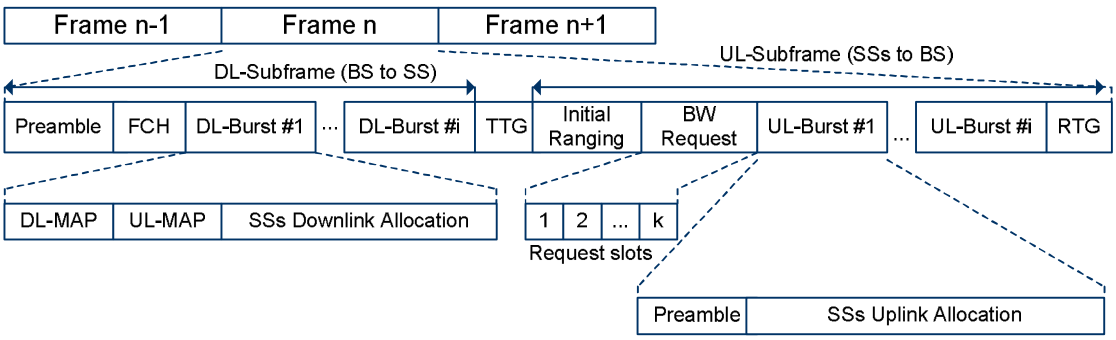


Figure 2.1: WiMAX frame structure

The communication link between the BS and the SSs consists of an uplink (UL) path (from SS to BS) and downlink (DL) path (from BS to SS). Uplink and downlink transmissions use frequency division duplexing (FDD) and/or time division duplexing (TDD) techniques. The WiMAX frame format consists of a DL sub-frame and a UL sub-frame (Figure 2.1). The DL path is a broadcast channel, sending the DL-sub-frame from the BS to all the SSs. The DL-sub-frame contains mapping information for both DL and UL transmissions. The main load is the DL data in the form of bursts and the

UL-sub-frame is sent from SS to BS. The UL path is a TDMA (time division multiple access) channel. The UL-sub-frame accommodates signal ranging, bandwidth request and UL data bursts. The SSs send data based on the allocated bandwidth in UL-MAP.

There are gaps between the UL and DL sub-frames; namely, the TTG and RTG. The TTG (transmit/receive transition gap) is a gap between the last DL burst and the first UL burst. TTG aims to provide sufficient time for the BS to switch from transmit to receive mode. The RTG (receive/transmit transition gap) provides time for the BS to switch from receive to transmit mode.

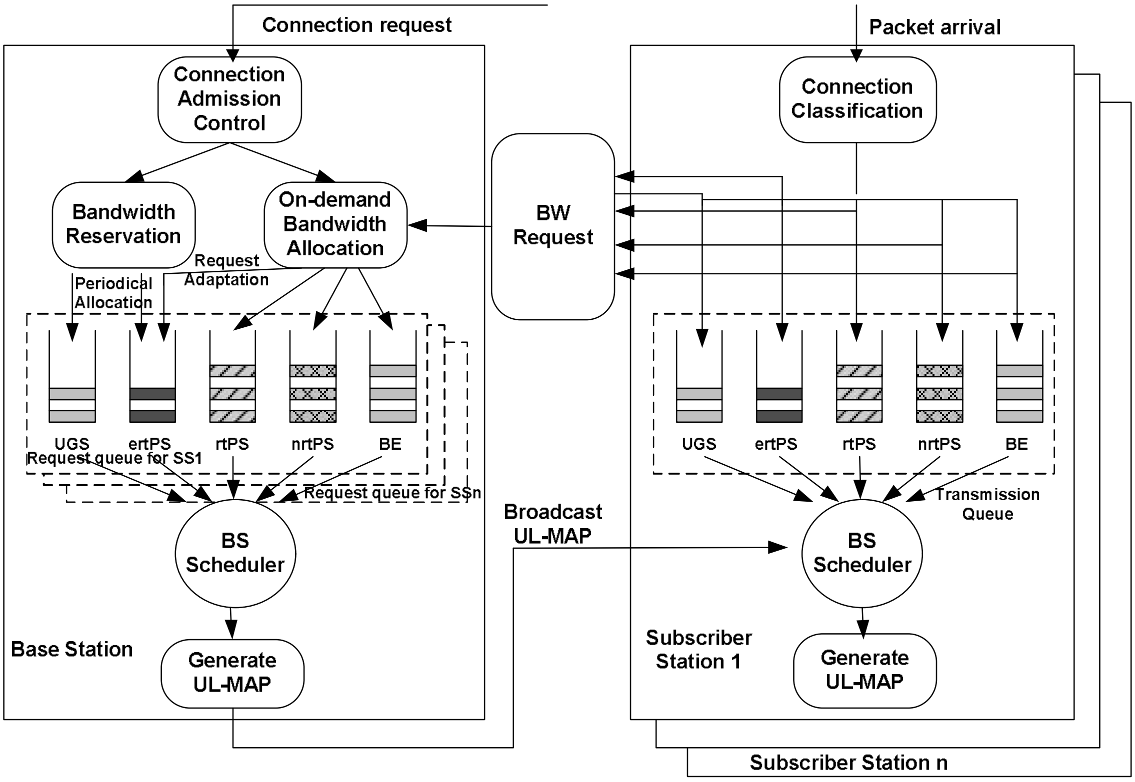


Figure 2.2: WiMAX MAC architecture [17]

Figure 2.2 shows the functional architecture of the WiMAX MAC layer. The architecture draws a logical connection between one BS and N number of SSs. A WiMAX SS requests bandwidth from BS by using a particular mechanism before sending data, depending upon the service type. The SS queues data in its MAC buffer, waiting for transmission. Since bandwidth request duration is limited, if SS misses the nearest bandwidth request opportunity, it should wait at least one frame period for another opportunity. The BS receives the SS bandwidth requests and processes them based on the type of the service. SS requests are then scheduled in the BS. Once bandwidth is granted by BS, SS receives notification in the UL-MAP of the next DL-sub-frame. The SS sends data in the following UL-sub-frame.

The scheduler in SS is important when there is more than one connection in one SS. Each connection requires an ID and its own bandwidth request. There are two bandwidth grant schemes in WiMAX; grant per connection (GPC) and grant per SS (GPSS). However, in the latest version of the standard, GPC has been omitted; therefore, the SS scheduler is often ignored as bandwidth is allocated to one SS. The scheduler in the BS is not defined in the standard and is left for vendor implementation. The BS scheduler and bandwidth request mechanism are designed according to user requirements. The following section provides details about MAC layer components.

a. Connection establishment

The SS should establish a connection with the BS before starting a conversation. Figure 2.3 depicts the connection establishment process of SS to enter a WiMAX network [18]. Initially, SS scans the DL channel to synchronize the air interface of the WiMAX link. If the working frequency and time slots are synchronized, SS obtains the network information from the DL-sub-frame. Then, SS contends to request initial ranging; SS should read the UL-MAP to adjust the power level. Once ranging is complete, the SS informs its capabilities to the BS and the BS shares its capabilities with the SS. Next, the SS authenticates its identity and registers with the BS. The SS obtains an IP address to the BS before making a connection. Periodical ranging is performed to ensure the SS has sufficient power to perform the transmission.

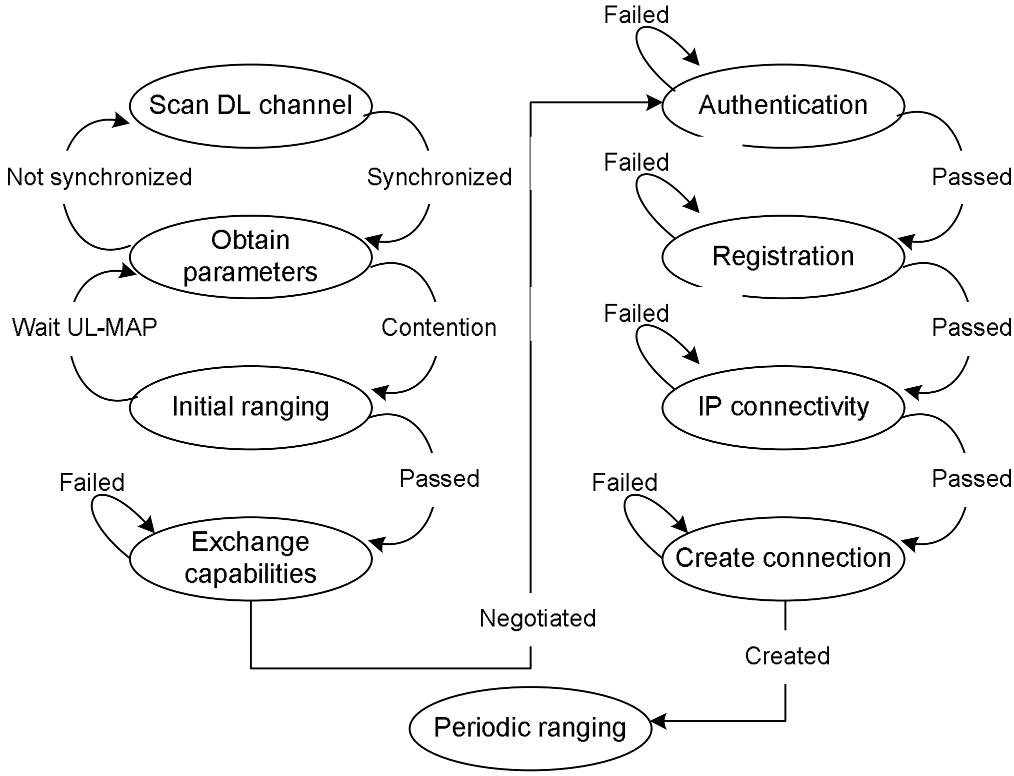


Figure 2.3: Connection establishment

The WiMAX standard defines three type of connections; basic, primary and secondary. Basic connection is intended for ranging and power control, as well as changing the data burst profile; primary connection is allocated to security management; and secondary connection is for data transfer and higher layer messages. A WiMAX connection is assigned to specific QoS parameters and a 16-bit connection ID (CID).

b. MAC frames

A MAC frame is a sequence of bytes that represents both the DL and UL sub-frames, and is also referred to as a MAC packet data unit (PDU), consisting of 6 bytes MAC header, maximum 2041 bytes payload and an optional 32 bits CRC (cyclic redundancy check). Only the bandwidth request message has no payload.

WiMAX MAC has fragmentation features. The longer packet, known as the service data unit (SDU), can be transported by several PDUs. There is additional allocation for the fragment header in PDU payload. Besides header insertion for SDU fragmentation, header insertion can also be used for piggybacked bandwidth requests. A brief explanation of bandwidth request is provided in Section d.

c. WiMAX services

The BS serves the SSs based on rules negotiated during the establishment of the connection. The type of service depends on the requested scheduling, which quantifies the QoS requirements used by the BS when allocating bandwidth. There are five types of scheduling services, as defined by the IEEE 802.16 standard:

- Unsolicited Grant Scheme (UGS) service has the highest priority. The service is designed to support constant bit rate (CBR) traffic that generates fixed size data in a constant period, such as T1/E1 connection and VoIP, without silence suppression. During connection establishment, the SS declares the required bandwidth, bandwidth allocation interval and maximum tolerable delay. BS allocates periodically the requested bandwidth.
- Real-time Polling Service (rtPS) supports real-time variable bit rate (VBR) service flow, such as MPEG video and VoIP, with silence suppression. The BS provides request opportunities to SSs with rtPS traffic by issuing polls periodically. The SS requests bandwidth without contending other SSs. Polling and request processing makes rtPS generating an additional delay in the bandwidth request-allocation process.
- Enhanced Real-time Polling Service (ertPS) combines the efficiency of the UGS and the rtPS classes. The typical application of this type is VoIP with silence suppression. The ertPS service allows SS to employ both reservation and on-demand bandwidth request and allocation.
- Non Real-time Polling Service (nrtPS) is used for non-real-time VBR application, such as the FTP application that requires a minimum bandwidth guarantee but can tolerate a longer delay. The nrtPS uses the same polling mechanism as rtPS; however, the SS is also allowed to contend for bandwidth request opportunity.
- Best Effort Service (BE) is intended for the best effort traffic that does not have any specific QoS requirements, such as telnet or http connections. The BE requires SS to request bandwidth.

d. Bandwidth request and bandwidth allocation scheme

Bandwidth request-allocation mechanisms manage and satisfy the UL bandwidth needs of the SSs. The mechanisms allow the SSs to indicate dynamically their bandwidth requirements.

The SS informs the BS of its bandwidth needs by sending a request message, which indicates the amount of bytes waiting for transmission at the SS queue. The standard supports different methods by which an SS can send the request to the BS. The SS requests bandwidth to the BS, either by a stand-alone BW-REQ message or a piggybacked request.

The stand-alone requests can be made using either unicast polling or a contention request. For the former, the BS allocates an opportunity for an SS to send its bandwidth request; while, for the latter, the SSs contend to send BW-REQ messages in the UL bandwidth request contention slots. The SS considers a request is lost if no data grant has been given in the UL-MAP within a certain time interval. The time interval is called T16 timer in the 802.16-2004 standard or contention-based reservation timeout in the 802.16e amendment. Bandwidth request losses are handled by a truncated binary exponential backoff (BEB) algorithm, which allows the retransmission of lost BW-REQs. The piggybacking method uses the grant management sub-header to attach a bandwidth request to a UL data packet.

The BS manages bandwidth perception to satisfy the UL bandwidth needs of the SSs. The perception management policies in the BS employ the algorithms that are implementation-dependent and not specified in the IEEE 802.16 standard. The BS optimizes the bandwidth requirement of the SSs by allocating the physical slots to the UL sub-frame and informing the SSs through UL-MAP, which is broadcast in the DL sub-frame.

2.3.2 Research on WiMAX

WiMAX has attracted significant interest from all subjects of wireless communications, including researchers, students, engineers and operators. Current WiMAX research involves optimization of components that are not defined in standard, handover issue, WiMAX coexistence with other networks, analysis of various components of the standard, security and other related issues.

The scheduling algorithms, which create perception policies to allocate WiMAX resources, are not defined in the standard. Research in [19] divided WiMAX scheduling into two categories; channel-unaware and channel-aware. The channel-unaware schedulers do not consider the channel condition for scheduling decisions, while the channel-aware schedulers exploit the channel characteristics of the requests. This thesis introduces the scheduler classification based on the traffic generated by the SSs: traffic content aware and traffic content un-aware schedulers. There are some schedulers that are not aware of the traffic content, such as Round Robin (RR); Weighted Round Robin (WRR); Deficit Round Robin (DRR); FIFO (First In First Out); Earliest Deadline First (EDF); and Weighted Fair Queuing (WFQ). Moreover, other schedulers are traffic-aware, such as frame-based [20] and priority-based EDF [21].The recognition of the traffic content is an improvement on the traffic-aware scheduler and Chapter 3 will explore this topic in more detail.

The call admission control (CAC) is also an open issue in WiMAX, as it determines the performance of the existing connections when a new one is admitted. The token bucket approach [22] and QoS hierarchy [23] are the example works on CAC.

The success of the mobility feature in WiMAX depends on the capability of performing fast, seamless handovers. Although the standard [13] has defined the MAC-layer handover management, which is expected to provide seamless handovers of fewer than 100 ms and almost zero packet loss for the SS speed of 120 km/h [24, 25], the path to commercialization of a full mobility is a research challenge that lies in the already standardized handover methods. Excessive scanning and association may occur when there is more than one neighboring BS [26], suspension of data exchange during scanning process [27] and disruption time during the handover [28].

Deploying two different systems in the same area using adjacent frequency bands leads to coexistence issues. Interferences introduce capacity degradation of both systems. The challenge posed by the coexistence issue is how to reduce the impact of frequency interferences. Coexistence research between WiMAX and other networks focuses on how to avoid frequency conflict and how to manage integration with existing technologies.

Coexistence WiMAX and WiFi occur mainly in unlicensed ISM frequency operations. Both systems have heavy interference risks, as both are designed to work in the same frequency [29]. Unlike WiFi, Code Division Multiple Access (CDMA) coexistence results in minor interferences, as CDMA and OFDM are completely different technologies [30]. Vertical handover and Joint Radio Resource Management (JRRM) have been proposed as a solution for overcoming the coexistence issue between WiMAX and HSPA [31]. Ultra wide band (UWB) frequency is often used for short-range radio devices, such as PC peripherals. A coexistence solution between WiMAX and UWB device is found by decreasing power level of the UWB device. Detection and Avoidance Algorithm (DAA) [32] and Spectrum Sensing Algorithm (SSA) [33] have been proposed. Since each technology experiences a different degree of development, research challenges on coexistence are wide ranging.

Many research challenges remain in relation to WiMAX and they are not limited to those described above. Internal challenges come from the development of the new WiMAX standard, while the external challenge is competition from technologies, such as 3GPP Long Term Evolution (LTE) [34].

2.4 Transport layer protocols

2.4.1 Unreliable transport layer protocol

The unreliable transport layer protocol in the TCP/IP stack is UDP. Delay-sensitive applications, such as real-time communication, networked games and streaming applications, use UDP as the transport layer protocol [35]. The characteristics of the UDP, which do not require retransmission and allow application-dependent data rate, make it suitable for those applications [36].

<table>
<tr><td colspan="3">Source address</td></tr>
<tr><td colspan="3">Destination address</td></tr>
<tr><td>Zero</td><td>Protocol</td><td>UDP Length</td></tr>
<tr><td colspan="2">Source port</td><td>Destination port</td></tr>
<tr><td colspan="2">Length</td><td>Checksum</td></tr>
</table>

Figure 2.4: UDP header, from [35]

Another reason to use UDP for delay-sensitive applications is low protocol overheads. Figure 2.4 presents the UDP headers. Shaded fields are provided by upper layer. The UDP header contains only source and destination ports, data length and checksum. The latter is optional; if the checksum is enabled, the packet is discarded if an error detected.

2.4.2 The need to improve the unreliable protocol

A wireless link is subject to signal interferences, noises and low signal strength, which can cause high packet loss. Since UDP does not provide retransmission for discarded and lost packets, the use of UDP for video transmission over radio link can cause an unacceptable ratio of lost packets.

A single packet loss can cause noticeable disruption in video rendering. Packet loss also can cause propagated error in video decoding. Therefore, there is a need to improve the unreliable protocol. Although the delay characteristic of UDP is desirable, packet loss in wireless transmission should be reduced. Chapter 4 of this thesis proposes improvements to the unreliable protocol by using an effective retransmission scheme and optimizing support from the lower layer.

2.5 Cross-layer techniques

The TCP/IP protocol stack defines separation layers for IP-based networks that work independently. Data are passed from one layer to another by using header encapsulation and de-capsulation. As shown in Figure 2.5, each layer adds headers to user data (application byte stream). The transport layer header and data form a TCP segment or UDP datagram. Additional network addressing headers create a network packet, while the MAC header creates a MAC frame.

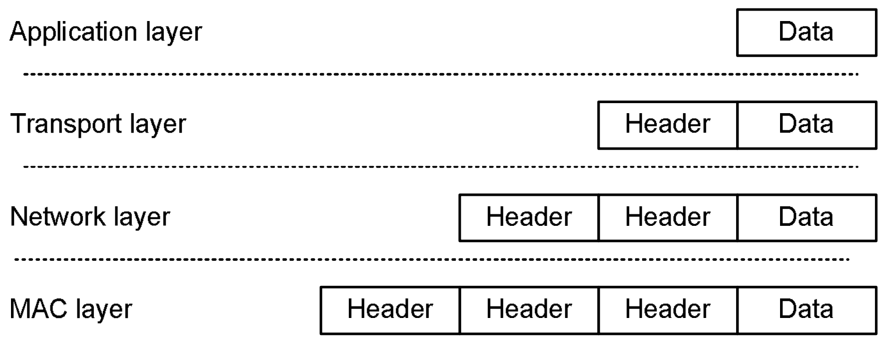


Figure 2.5: TCP/IP data encapsulation

Although each header passes to lower layer, upper layer headers are not readable to lower layer entities, because they are intended for the same layer entities at other communication endpoints. Only the same layer entity can read information in the header; each layer is completely separate. This idea of layer separation may work well in wired media, as the communication channel is not prone to channel condition. However, layer separation performs poorly in a wireless environment, as device characteristics and channel quality vary quite often [37, 38].

The cross-layer approach is considered a proper solution for wireless communication as it enables each layer to interact by passing information uni-directionally or bi-directionally through separate PDU or existing PDU headers. Upper layer headers may be set as readable by the lower entity at the same endpoint. Raisinghani and Iyer [37] outlined a cross-layer possibility in different layers.

Chapter 3 of this thesis discusses the cross-layer scheme between MAC and transport layers. The cross-layer scheme is intended to support the designed transport layer protocol.

2.6 Application layer techniques

Research on improving multimedia streaming is not limited solely to the lower level of network stack, but also to the application layer. Vandalore et al. [39] classified application layer techniques into three levels: encoding, streaming and operating system.

Encoding refers to the format of the streamed video and the quality of the streamed data, and deals with video compression techniques. There are discrete cosine transformation (DCT), wavelet encoding, and proprietary compression methods. The quality of video is determined by adjusting the frame rate or video resolution.

The streaming techniques are performed in two ways; passive or reactive. The passive method optimizes the usage of network, while reactive methods modify traffic (such as encoding properties) to suit the network.

The passive method can use either layered encoding or redundancy. Layered encoding translates video data into several layers. The base layer has minimal video information, and additional layers add quality to the video. The more the layer is sent, the higher network bandwidth is required and the higher video quality is received. The redundancy method uses additional data with the video data; for example, Forward Error Correction (FEC), which sends checksum redundancy to minimize the error rate.

Rate shaping is an example of the reactive method that uses a feedback mechanism to decide which quality is best for streaming data. The buffer occupancy level is the network feedback source. This feedback is used to change the streamed data in the passive method.

Chapter 5 of this thesis discusses the implementation of rateless code and error concealment techniques for the application layer improvement of WiMAX for dedicated video surveillance network. Both rateless code and the proposed error concealment techniques use passive redundancy.

Operating system (OS) level techniques may not be needed as the performance of computer systems has increased significantly, especially in processor speed and memory capacity. The OS scheduler, which assigns priority to multimedia applications, is an example of the OS method.

2.7 WiMAX simulation

Simulation is an appropriate method of examining the performance of a system. Simulation avoids costly assessment in a real system and shortens the time for obtaining the final results. Implementing the proposed adjustment of MAC layer on the real WiMAX device could be very expensive, as researchers should have access to device firmware as part of the manufacturer's copyright. Simulation is the right choice to represent the system by employing some assumptions from the original, without reducing the accuracy of the results.

There are many WiMAX simulators available, either open source or commercialized. Opnet, Qualnet and Matlab are examples of commercial simulators. The open source simulators, such as NS-2 and omnet++, are alternatives that have similar capabilities to the commercialized ones.

This thesis uses NS-2, which is used widely to simulate wireless networks. The decision to use NS-2 is motivated by the easiness of packet header modification and monitoring for performance evaluation purpose. The NS-2 simulator used in this thesis is version

2.34, accompanied by the external WiMAX module developed by the NIST [40], channel and error models, and an Evalvid evaluation framework.

2.7.1 NS-2 simulator

The NS-2 simulator, a discrete-event network simulator was developed by the University of California, Berkeley and maintained by the University of Southern California. NS-2 uses two languages: C++ and OTcl (object oriented extension of Tcl language). The object-oriented part of the simulator is written in C++ for fast and efficient execution, while OTcl is used for configuration, such as network structure and topology.

The simulator is developed for research and education in protocol and network design. NS-2 was intended originally for wired networking, but is now used widely to simulate wireless networks [41]. NS-2 provides various protocol modules, network and traffic types.

Since NS-2 does not provide a WiMAX module in its package, some researchers and institutes offer WiMAX modules, such as the National Institute of Standards and Technology (NIST) [40] and the Network and Distributed Systems Laboratory (NDSL) [42]. The NIST and NDSL modules offer partial fulfilment of the standard. The NIST WiMAX module uses OFDM (Orthogonal Frequency Division Multiplexing) scheme with mobility, fragmentation supports. Conversely, the NDSL WiMAX uses OFDMA (Orthogonal Frequency Division Multiple Access) scheme with CAC support. Since this thesis considers the mobility of the subscriber stations, the NS-2 simulator is patched with the NIST WiMAX module.

2.7.2 NIST WiMAX Module

The NIST module was built according to the WirelessMAN-OFDM with configurable modulation. The UL and DL connections are separated by Time Division duplexing (TDD). The module provides spaces for authentication and scheduling, so that researchers can implement their research by modifying these spaces. By default, there is no authentication in the network entry stage and round robin is the basic UL schedulers. The module also supports scanning and handovers, as well as fragmentation and frame reassembling.

Figure 2.6 highlights the structure of the NIST WiMAX module. The primary Mac802_16 module extends the MAC module within the NS-2 simulator. There are six major components of the module: peer node; connection; service flow; classifier; scheduler and statistics. Peer node module records information about peers, including the SSs and BS.

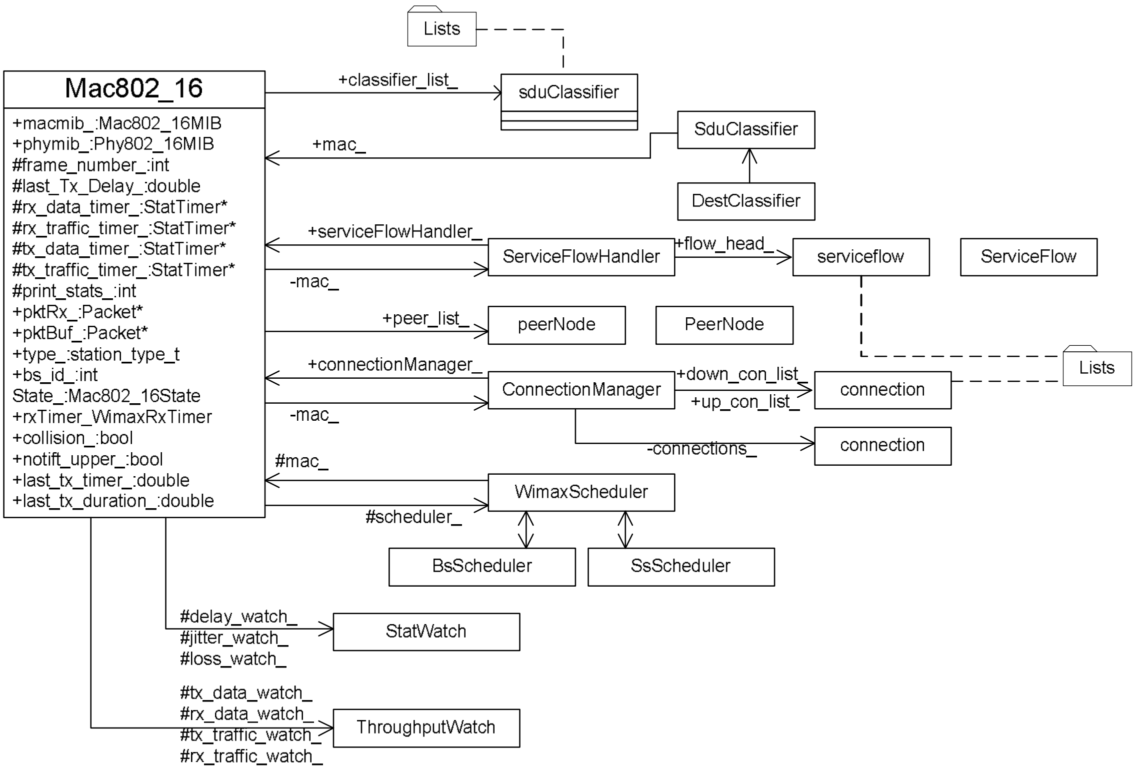


Figure 2.6: The NIST WiMAX module[40]

Since WiMAX uses the grant per SS scheme, each peer has only one connection. The properties of the incoming and outgoing connections are managed by connection module. Each connection can contain several service flows, which are handled by the service flow module. The classifier module records and processes the incoming and outgoing packets. The scheduling algorithm is implemented in three scheduler modules.

The NIST WiMAX module and the enhancements added to NS-2 can be validated in several ways to test and verify the correctness of the added functions and the compliance to the IEEE 802.16 standard. Some of the validation methods are link adaptation, data rate validation, frame validation and QoS validation. The link adaptation is to validate the correct trend of the Signal to Noise Ratio based on SS positions; the data rate validation measures the cell bandwidth consistency; the frame validation is to check the frame format in TDD mode; and the QoS validation to check the correctness of each class of service. The NIST module validation has been presented in many works [142-144], this thesis uses data rate validation to ensure the surveillance system works in a non-saturated network.

2.7.3 Channel model

Channel models are used extensively in site planning, particularly for wireless coverage feasibility studies and network adjustment. The channel model in the network simulator is also required in order to approximate signal propagation. The channel models can be categorized as: empirical, deterministic and stochastic [43]. Empirical models are based on observations and measurements, and the empirical channel models are usually used to predict the path loss. The deterministic channel models use electromagnetic wave analysis to determine signal power reception. Stochastic channel models approximate the channel as a series of random variables.

Among the models, the empirical models require the most extensive data. The study on specific channels usually makes use of the most likely propagation model of the available propagation data. Once the referred model is available, the empirical model yields the most valid results. The example of empirical models is the two-ray propagation model [44], the Stanford University Interim (SUI) [45] and the COST-231 Hata model [46].

The deterministic models intensively use mathematical formulae and often require a complete 3D representation of the environment. However, the basic deterministic models, such as the ray tracing model [47], can provide a quick propagation model.

As most surveillance applications are implemented in a well-known communication link, this thesis uses a two-ray propagation model for a good propagation environment and COST-231 model for an erroneous environment.

2.7.4 Error model

The channel models adjust simulations to achieve a realistic environment model. However, the final result of the channel models is only received signal strength. A simulator cannot determine whether or not the received signal level contains error;

therefore, an error model is required when simulating wireless networks with erroneous channels.

This thesis uses a 2-state Markov model in the NS-2 simulator for surveillance application in an erroneous environment. This model tracks the received signal in either good (G) or bad (B) states, either of which may generate errors. If (1 - k) and (1 - h) are dependent error rates in good and bad states respectively, p is the probability of error transmission providing the previous transmission was successful, and r is the probability of successful transmission if the previous one was in error. The 2-state Markov model is illustrated in Figure 2.7.

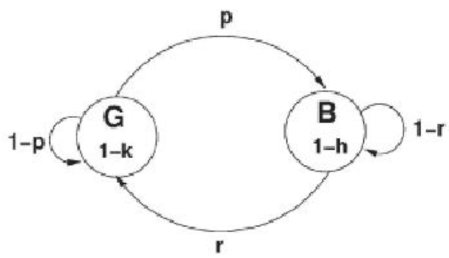


Figure 2.7: The 2-state Markov model

The error rate P_E is determined by the stationary state probability π_G and π_B, which exist if p>0 and r < 1, as given by [48]:

$$P_E = (1 - k)\pi_G + (1 - h)\pi_B \tag{2.1}$$

$$\pi_G = \frac{r}{p + r} \tag{2.2}$$

$$\pi_B = \frac{p}{p + r} \tag{2.3}$$

In order to obtain the error rate, parameters p, r, k and h should be solved. There are some approaches to obtain these values. This thesis employs the simple Gilbert model, which assumes a good signal produces successful transmission (k=1) and a bad signal causes errors (h=0). The error rate is set initially to 0 to reflect the best transmission environment, and set to 0.1 to represent an error-prone environment.

2.7.5 Evaluation framework

In order to evaluate the received video in NS-2 simulations, this thesis uses the Evalvid evaluation framework. Evalvid is a tool-set for evaluating video quality transmitted over a real or simulated communication network [49]. Figure 2.8 depicts the framework.

A video source file is encoded and compressed by the video encoder. The video source used in Evalvid can be either in the YUV CIF (352 x 288) or YUV QCIF (176 x 144) format. The video encoder and decoder use the NCTU codec [8] and ffmpeg [9]. The video sender (VS) reads the compressed video file and transmits it over a real or simulated network. The VS also generates a video trace file for the Evaluate Trace (ET). The video packets are received at the Play-out Buffer and decoded into a YUV format.

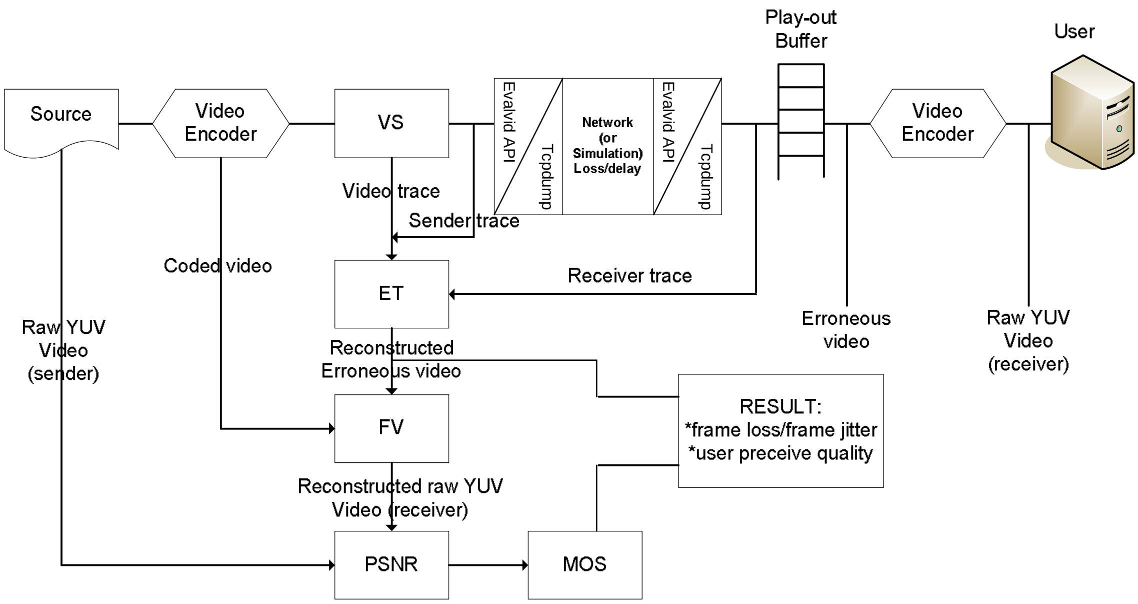


Figure 2.8: The Evalvid framework

By comparing the trace files, ET creates reports of loss and jitter. The fix video (FV) substitutes the lost parts of the received video to produce a video reference for comparison purposes. The evaluation parameters are then computed to express the received video performance.

In order to implement the Evalvid framework in the NS-2 simulator, the approach that was described in [50] is used. Ke, Lin and Shieh [50] added three connecting interfaces to the NS-2 simulator, namely MyTrafficTrace, MyUDP and MyUDPSink, so that the trace can be generated by NS-2. The Evalvid implementation in NS-2 can be seen in Figure 2.9.

MyTrafficTrace reads the video file, converts video frames into smaller video packets and then sends the packets to the transport layer. The MyUDP module is the extension of the transport layer module, which handles host-to-host communication. MyUDP records the packet sent to the other point. In contrast, MyUDPSink receives the sent packets and records the packet sequence, size and receiving time in the trace file. Finally, the ET module of the Evalvid framework computes the performance of the generated trace files.

The transport layer part of this thesis modifies MyUDP and MyUDPSink modules in order to accommodate the proposed techniques.

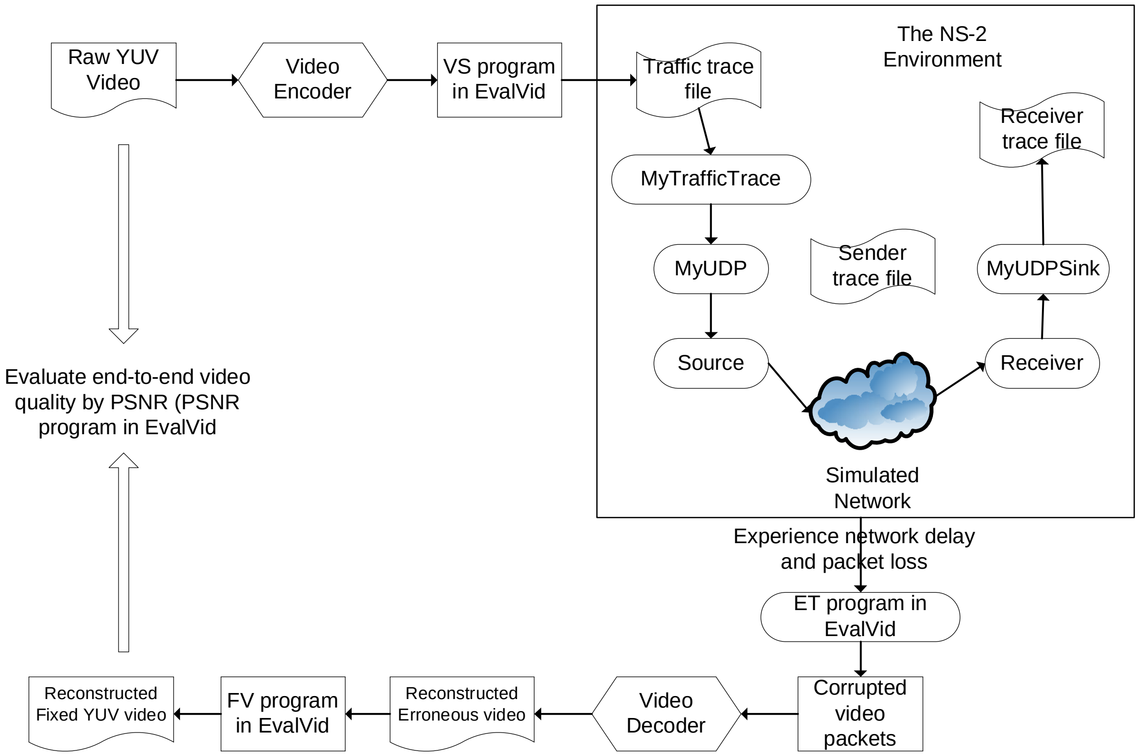


Figure 2.9: The Evalvid framework

2.7.6 Evaluation metrics

As this thesis proposes multi-layer improvements on video surveillance applications over WiMAX, performance metrics are required to assess the level of improvement relative to the existing methods.

The video evaluation metric is required to measure video degradation caused by media transmission or processing system. The measurement can be performed objectively or subjectively. The widely-used objective video quality evaluation is peak signal to noise ratio (PSNR). Subjective video evaluation estimates viewer opinion through quality scaling; for example, minimum opinion score (MOS) [51], which is not used in this thesis.

Although other metrics can be found throughout this thesis, the main performance metrics are delay and PSNR. Packet delay is the time difference between packet transmission and packet reception. PSNR is a well-known objective metric for assessing the application-level quality of video transmissions. The PSNR is computed as the difference of the luminance component Y of source and destination images:

$$PSNR\ (dB) = 10log_{10}\frac{r^2}{\sum_{M,N}[I_1(m,n) - I_2(m,n)]\big/_{MxN}} \tag{2.4}$$

where r is the maximum image fluctuation, MxN is the image size; I_1 and I_2 are points of the compared images.

In some sections, packet loss is presented instead of using PSNR. This is applied to large video transmissions, as PSNR comparison for long video duration in Evalvid evaluation framework is not realistic. This thesis also uses additional metrics for explanation purposes; for example, interface queue (IFQ) and requested bandwidth. Those metrics are discussed in subsequent sections.

2.7.7 Simulation scenario

The simulation scenario in this section is applied to the chapters using the NS-2 simulator to evaluate the performance of the WiMAX network.

The simulation scenario uses a basic PMP system where the WiMAX network has a single BS with several SSs communicating with it. The number of SSs is set to 4 as system capacity is limited. The Each node had a different speed to represent some possible surveillance positions. Node 0 was fixed (0 m/s); Node 1 was set to a walking speed of 1.39 m/s; Nodes 2 and 3 were assumed to be in a public transportation vehicle, such as a bus or tram; Node 2 moved at 4.44 m/s; and the speed of Node 3 was 6.67 m/s. The single cell network configuration is shown in Figure 2.10.

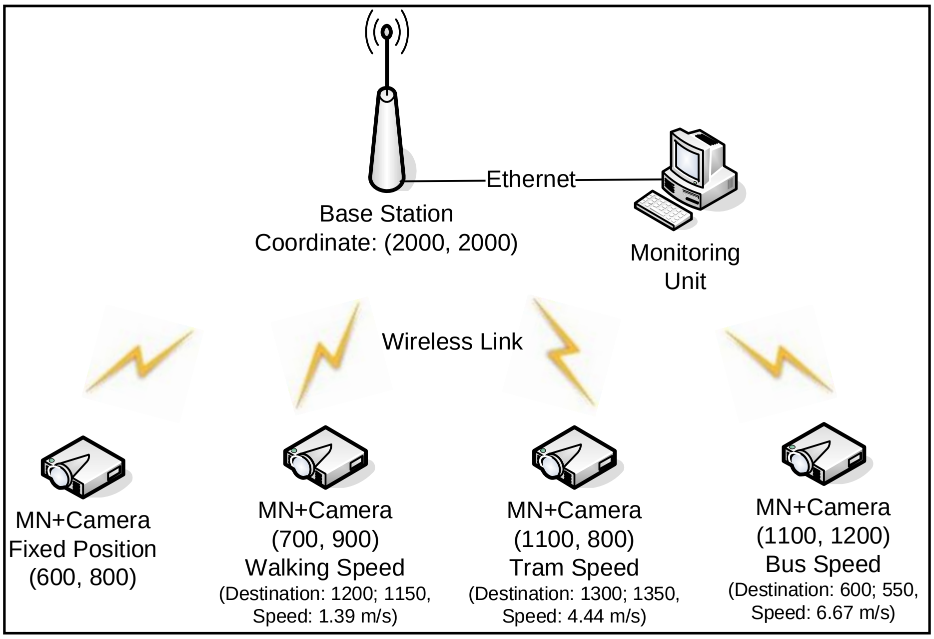


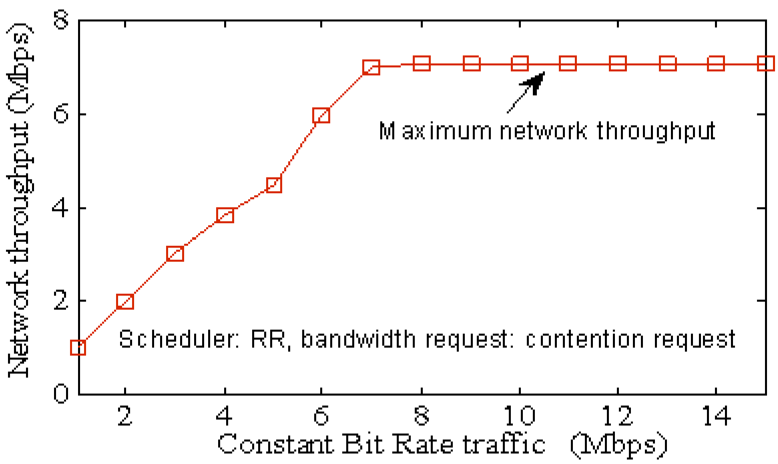
Figure 2.10: Network configuration

The WiMAX transmitter power and receiver threshold were set to provide a 1000 m coverage radius. The modulation scheme was 64 QAM. When the surveillance system applied in non-error prone environment, a two-ray ground propagation model was used, but when the link is erroneous, the Hatta model is selected.

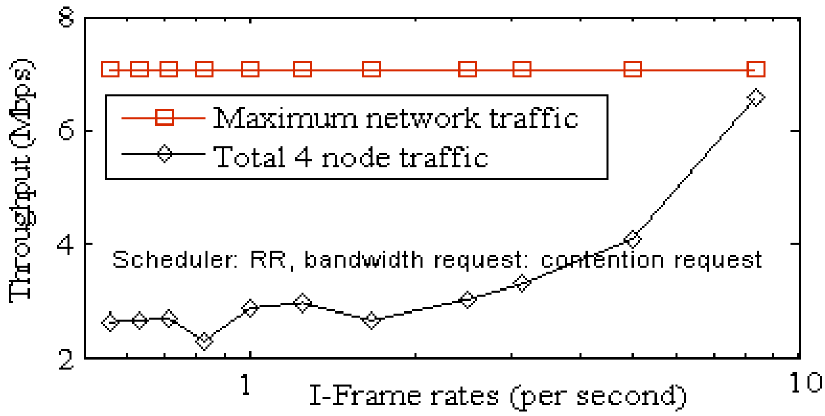
The DL to UL ratio was set to 0.3, which means that 30% of the network resources were allocated to DL. The simulated surveillance application had 4 mobile nodes (MN) within one base station. The number of mobile nodes was chosen to simulate a non-saturated network, which means that the traffic load was smaller than the network resources. This is important as a surveillance network should provide sufficient bandwidth in order to maintain video quality.

By using constant bit rate (CBR) tests from 1 to 15 Mbps, the saturated UL bandwidth is obtained and presented in Figure 2.11a. The network can accommodate total UL traffic up to 7Mbps. When total traffic exceeds this figure, the network throughput is saturated at 7 Mbps, which means that maximum network throughput is 7Mbps.

Since the proposed methods deal with packet/frame types, the traffic load is increased based on I-frame rates instead of the number of SSs. The total video rate for each simulation is depicted in Figure 2.11b.



a. Network throughput



b. Total network traffic

Figure 2.11: Network and total traffic throughputs

The traffic sources were generated from akiyo_cif.yuv video, a YUV CIF video file taken from [52]. Its video trace was used as simulated traffic in the NS-2 simulations, where the received patterns were reconstructed based on the original video. The traffic generation, reconstruction and evaluation in the NS-2 simulator were based on the Evalvid video evaluation framework from [49, 50], as discussed in Section 2.6.

When frame priority was applied, the prioritized frames were set for I-frames. All the packets generated from these prioritized frames were assigned as prioritized packets. Table 2.1 summarizes the traffic load parameters.

Table 2.1: Simulated traffic parameters

Parameter	Value
Video sequence	akiyo_cif.yuv
Frame rate	30fps
Frame type	IPP
Video codec	MPEG4
Video bit rate	559.35 Kbps for GOP 30
Packet size	1024 bytes
Group of Pictures	3, 5, 8, 10, 15, 20, 25, 30, 35, 40, 45

The performance evaluation was conducted by observing the sending and receiving of ports in each connection. The measurement in the NS-2 simulator refers to those in [50].The main performance metrics are the average delay and PSNR of the 4 nodes. Measurement points are in SSs (sender) and in monitoring unit (receiver). The PSNR is obtained by reconstructing the video from the received packets and comparing it with the original source

2.8 Summary

This chapter describes the layers involved in the performance enhancement of the WiMAX-based surveillance network, in addition to the simulation components used in performance evaluation.

WiMAX MAC architecture, services, bandwidth allocation and associated research are highlighted briefly to give an overview the opportunity of MAC performance enhancement in WiMAX-based video surveillance. In transport layer, UDP advantages, drawbacks and the need to improve it are summarized. UDP is used as the base of the proposed protocol in WiMAX-based video surveillance. Classification of the application layer techniques is presented as a direction in which the work is performed. The cross-layer techniques as the multiple layer coordination are also presented.

Finally, this chapter explains the NS-2 simulator, WiMAX module, traffic sources, channel and error models that are used as the simulation evaluations in most of this thesis.

Chapter 3
Related works

3.1 WiMAX for surveillance network

Fixed surveillance applications have been used widely to provide activity and security monitoring. However, mobile surveillance application is still restricted by the speed limitation [53]. WiMAX technology has attracted researchers to study its technology and its implementation, including for surveillance application. Although WiMAX-based surveillance implementation remains limited, the technology has potential to be used as surveillance infrastructures.

The research on the use of WiMAX for surveillance application work in two areas: implementation and performance enhancement. The surveillance implementation researches are performed either by using real devices or through simulations. Generally, the works assess the possible surveillance topologies, interaction among surveillance components, as well as the video surveillance quality assessment. Chang et al. [54] studied WiMAX-based surveillance application for the disaster-field command system. The work was performed in the lab using real devices. Their research integrated the mobile sub-system with the command center. Besides using the full network, WiMAX can be combined with other networks, such as ethernet. Aguado et al. [55] analyzed and simulated the implementation of WiMAX to connect with an ethernet-based CCTV network in a moving train to the monitoring center.

The problem with the study in terms of implementing WiMAX-based surveillance in a real device is that there are limited opportunities to modify physical and MAC layers as access to the device is restricted. Therefore, this thesis does not implement lower layer modification on experimental implementation.

The WiMAX-based surveillance performance enhancements are conducted in 4 different layers: MAC; network; transport; and application layers. The network layer enhancement is performed when the assessed surveillance network is vast or its topology is a mesh network. For instance, Ababneh and Rougier [56] considered a mesh surveillance network in their study and proposed a routing scheme. This thesis considers that network layer enhancement is not significant, as most surveillance networks are centralized and mesh topology is rare.

There are various researches on MAC, transport and application layer improvements on WiMAX. However, only a few were focused on WiMAX-based surveillance. FEC adjustment [53], WiMAX-based surveillance using SIP (Session Initiation Protocol) [56],and adaptive coding for video surveillance [57] are examples researches on MAC, transport and application layers respectively.

3.2 WiMAX bandwidth request mechanisms

Existing literatures discussed bandwidth request in WiMAX in two forms: modeling the mechanisms and proposing the enhancements. Research in [2, 61-67] presented mathematical models for bandwidth request mechanisms. There are various bandwidth request models, such as the slotted Aloha-based model [58]; Queue-based model [59]; Calculus-based model [60], Control theory-based model [17]; and Markov Chain-based models [2, 61-67]. Fallah et al. [61] concluded that the slotted Aloha scheme cannot accurately model the 802.16 contention access. The Queue-based models were used for polling-based bandwidth requests; the Calculus-based method is derived from the Queue-based model; the Control theory-based models consider the stability factor and the Markov Chain based models are the most frequently-used models in 802.16 analysis. The models proposed in [2, 61-64] are intended for contention request, the work in [59] is for unicast or polling based request, and literature [65] discussed both bandwidth request mechanisms. In those models, the analysis was performed either in saturated or non-saturated network conditions. Vu, Chan and Andrew [2] emphasized that a saturated condition is important for understanding upper-bound performance. However, Ni, Hu and Vinel [63] considered that networks typically do not operate in saturated conditions.

Besides contention and unicast requests, there is another mechanism known as piggybacking bandwidth request. Literature [66, 68] detailed the bandwidth request mechanism. Contention request with piggyback [66] is a method that rides alongside the bandwidth request for the remaining data into data burst if the allocated bandwidth is not sufficient to carry all data available in the queue. For instance, if the requested bandwidth is 1500 bytes but the allocated bandwidth only covers 1000 bytes, the bandwidth request for the remaining 500 bytes is piggybacked into data burst. A contention request with piggyback is aimed at a single frame backoff [66], which means only one contention request for the separated data bursts. He et al. [67] proposed an analytical model for a contention request with piggyback. Results from [66, 67] show that a contention request with piggyback outperforms the standard contention request.

The previous paragraphs focus on bandwidth request modeling. The following discussion summarizes briefly the existing enhancement on bandwidth request mechanisms in the standard. The improvements to the existing contention-based bandwidth request emphasized the modification of the truncated binary exponential backoff (TBEB). Kwak et al. [68] proposed an exponential increase exponential decrease (EIED)-based contention resolution mechanism for ranging in the WiMAX network. The EIED was used initially in 802.11 networks [69]. The objective of the EIED backoff algorithm is to minimize the collision probability by randomizing the transmission timing. The contention window (CW) size is adjusted dynamically depending on the collision history; increasing whenever a collision occurs, and decreasing when transmission is successful. The idea was improved by Rajesh and Nakkeeran [70], who enhanced EIED backoff with multi-stage contention resolution (MSCR) for the WiMAX network. The MSCR reduces the overlapping probability of backoff counters among stations. The Utility Based Backoff (UBB) Algorithm was proposed by Thapa and Shin [71] for initial ranging in the WiBro network. In UBB, instead of using an exponential increment, the CW increment is the function of satisfaction utility of the SS on its deferred backoff value on the previous state. The higher the backoff value in a previous state, the lower the range of CW size and vice versa. The method proposed by Chou et al. [72] simply adjusted time dynamically out of the backoff algorithm to achieve a better contention request.

Improvements to unicast bandwidth requests have been proposed in some literatures [73-75] for different applications. Mukul et al. [73] proposed a capacity increment on current bandwidth request, which is allocated for the next rtPS traffic. The method performs well when bandwidth is overwhelming; however, the method potentially reduces network performance as the additional bandwidth may be wasted. Liu and Chen [74] proposed an adaptive bandwidth request by adjusting the transmission sequence of the polled traffics. Although the authors claimed the method performed better than the original scheme, it requires a major change to the standard as the slots must be rearranged into contention free and contention period slots. Nie et al. [75] proposed an adaptive polling service (aPS) that considers the ON/OFF mode of the traffics. During ON periods, polling intervals are fixed and short, while OFF periods polling intervals are lengthened exponentially. The method depends heavily on the precision of the ON/OFF period of the traffic. Although it also proposed piggybacking, the method may produce excessive unloaded traffic. Park [17] proposed a simple and efficient UL bandwidth request algorithm for the ertPS scheduling mechanism; however, it was intended for SS side. Lee et al [76] proposed the CDMA code-based bandwidth request scheme, but time out and code adjustment require processing time that is not suitable for uniform real-time video traffic.

Pries, Staehle and Marsico [77] proposed and analyzed the performance of contention request, which piggybacks the bandwidth request for the next incoming data into the current data burst. Such piggybacking is appropriate when the traffic has a constant rate, so that the incoming number of bytes is known. Otherwise, the number of requested bytes in a piggyback should be predicted. The latter scheme is called as 'next frame piggyback'.

3.3 WiMAX scheduling algorithms

Scheduling algorithms are implementation-dependent and not specified in the standard. The basic legacy scheduler is Round Robin (RR), which examines and allocates bandwidth requests sequentially. The Weighted Round Robin (WRR) and Deficit Round Robin (DRR) modify the RR scheduler by applying different weights that represent node selection frequency. FIFO or FCFS (First Come First Served) prioritizes services based on the earliest arrival time. While the EDF, a well-known scheduling algorithm, prioritizes a node with the earliest deadline, Weighted Fair Queuing (WGQ) uses separate FIFO queues and processes non-empty queues simultaneously. Sophisticated schedulers, such as EDF and WFQ, may not work properly in dedicated surveillance networks as the schedulers work according to different priorities for different traffics, while in dedicated surveillance networks, traffic is uniform.

Dhrona et al. [78] evaluated various scheduling algorithms for UL traffic in the WiMAX network. Among the schedulers, Hybrid EDF, WFQ and FIFO produce the highest throughput. Hybrid schedulers, which employ multiple legacy schemes, have also been evaluated in [79, 80]. They perform better than the legacy algorithms as they satisfy the QoS requirements of the multi-class traffic. Each scheduler serves a different traffic class in a strict priority manner; for instance, the hybrid EDF, WFQ, and FIFO in [78, 79] employ the EDF scheduler for rtPS service, WFQ scheduler for nrtPS service and FIFO scheduler for BE service. In a dedicated surveillance environment, where traffic is video, the hybrid scheme is not appropriate.

Noordin and Markarian [19] classified schedulers into two types: channel-unaware and channel-aware schedulers. Channel-unaware schedulers assume that physical properties are stable. The paper also proposed a strict priority scheduler with minimum bandwidth allocation to avoid bandwidth starving. The scheduler is channel-unaware, but considers indirectly channel quality. Since its priorities are set for different service classes, the scheduler is not suitable for uniform traffic. Literatures [81-83] include other examples of channel-unaware schedulers.

A cross-layer scheduler, in contrast, is a channel-aware scheduler. The cross-layer scheduler in [84] obtains parameters from another layer and adjusts them within the current layer. The cross-layer algorithm in [85] employed a priority function at MAC layer and a slot allocation policy at physical layer. The scheduler used connection history in the allocation decision. It reallocates the slots from the most satisfied user to the most unsatisfied user. Although the scheduler increased fairness, the system capacity decreased. The scheduler in [86] has four scheduling stages. In the first stage, the priority queue allocates directly UGS traffic; the remaining classes are rescheduled using WFQ. The weight is calculated in accordance with priority and channel quality. The remaining bandwidth is prioritized to the latency-sensitive BE. If there is more than one BE traffic, RR is employed. Although the author claimed that the scheduler achieves both system capacity and fairness, the multi-stage scheduler imposes a higher delay on BE traffic. Literature [87] proposed a cross-layer scheduling algorithm for multiple connections with diverse QoS requirements, where Adaptive Modulation and Coding (AMC) and Genetic Algorithm schemes were employed depending on the quality of the wireless channel. The scheduler increases system capacity, but does not consider the delay.

Despite the system capacity increment offered by the cross-layer schedulers, the processing time is increased. Since all surveillance nodes intensively send video data, cross-layer schedulers may impose a high processing delay. The dedicated surveillance network is expected to have proper network planning; therefore, poor channel quality in one node may not affect other nodes.

There are many schedulers designed for various WiMAX-based applications, such as in literatures [88-93]. Works [88, 89] proposed schedulers for voice-over internet protocol (VoIP) application over WiMAX. Brahmia et al. [90] and Wang et al. [91] provided scheduler design for internet protocol-based television (IPTV). Sabri et al. [92] modified rtPS by reducing adaptively the polling interval to improve VBR performance in UL WiMAX. Although the proposed method improves video streaming performance, it is still bound to the unicast bandwidth request. Wu et al. [93] investigated scheduling for multiple connections in a single SS. The scheduler was designed for SS.

In video streaming, packet types become important as video codec generates frames with different priorities. Packet-aware scheduling for video traffic was introduced in [20, 21]. Kang and Zakhor [20] proposed a scheduling algorithm based on an unequal deadline threshold for wireless video streaming (frame-based scheduling). The SS scheduler increases the deadline from 0 (I-frame) to maximum value (P-frame immediately before I-frame). The frame-based scheduling performs better than EDF for video transmission as I-frame is prioritized. Wang and Liu [21] proposed priority-based EDF, which modified the deadline requirement based on frame type. Basically, the work is similar to [20], except that it considers other traffic classes. Both methods perform similarly in uniform video surveillance, as only one traffic class is involved. The frame and priority-based EDF are sorting schedulers.

3.4 Unreliable transport layer protocol

In early video streaming applications, content is transferred via two modes: point-to-point and point-to-multipoint. One of the most popular video streaming applications using both modes is video traffic transmission for television (TV). In this application, the TV station sends video traffic to the relay station through a leased line channel using either wired or wireless technologies. The relay station broadcasts video in point-to-multipoint mode to the TV receivers. In the rapid development of internetworking, online video application has become a popular application. Besides point-to-point and point-to-multipoint modes, the increasing demands on file sharing enhance the popularity of peer-to-peer systems. The scalability and accessibility of these systems motivated research on peer-to-peer streaming applications [94]. Rapid video application developments have generated interest in exploring approaches to increase live video streaming performance [95]. One way is to improve the transport layer protocol [96].

Traditionally, video streaming uses UDP as the transport protocol [35]. UDP does not perform retransmission and error recovery, both of which are attractive for delay sensitive applications. However, a wireless channel and physical hardware are characterized as bandwidth limited and unreliable, which induce packet losses as a result of channel interferences and network congestion.

In order to improve video streaming performance, UDP should be enhanced to reduce packet loss rate. UDP improvement has been proposed in many ways, such as by adding TCP properties to UDP [97-99], utilizing CRC (Cyclic Redundancy Check) within UDP packets [36, 96], or applying NACK-based retransmission [101-102]. Adding TCP properties could reduce the congestion problem in UDP; however, it could change the nature of the unreliable protocol. For example, RUDP (Reliable UDP) [97], which uses congestion control mechanism, acknowledgement and re-transmission services, often experiences delays as high as TCP [103]. DCCP (Datagram Congestion Control Protocol) [98] implements two types of congestion control; TCP-like and TFRC (TCP Friendly Rate Control)-like. DCCP relies on client feedback to perform congestion control. Loss of feedback packets may reduce DCCP throughput as the DCCP Sender assumes packets are not received and the sender adjusts the sending rate to half. Misperception of DCCP rate control can result in the underutilization of networks.

Conversely, CRC-based protocols, such as UDP-lite [36], may pass unacceptable error packets to the upper layer. Although it preserves the unreliability of UDP, UDP-lite is not compatible with traditional UDP applications, and error packets may exert problem in the application layer [104]. Another CRC-based improvement is CUDP (Complete UDP) [96], which integrates error information in the link and transport layers. However, the protocol is bound to the link layer and is sensitive to network congestion.

Although retransmission was abandoned in CRC-based improvement, NACK-based protocols, such as BVS (Broadband Video Streaming) [102], are compatible with existing UDP applications. Moreover, video streaming applications are frame-dependent, which means the frame decoding may require other frame(s) to decode; therefore, retransmission delay could be neglected. The main problem of existing NACK-based protocols is that not all are designed for video transmission. Only BVS is intended for media streaming; however, this protocol uses multiple retransmission requests for separated packet loss in one video frame, thereby rendering retransmission ineffective.

There are few existing works on transport layer protocols that employ NACK-based retransmission in various applications. In high speed networks, UDT (UDP-based Data Transfer) [100] and RBUDP (Reliable Blast UDP) [101] were proposed based on UDP. Both protocols aimed to solve the TCP weakness that underutilizes high-speed networks. UDT employs both ACK and NACK packets. The ACK packet carries information regarding reception speed from the receiver. The sender keeps increasing the transfer rate until it receives the NACK packet, which communicates that packet loss has occurred. The sender then decreases its transfer rate. In contrast, RBUDP employs only NACK packet, which sends a TCP request-reply to acknowledge lost packets in a UDP-based bulk transfer. If UDT schedules a NACK packet to be sent as soon as possible when packet loss is detected, RBUDP sends a NACK packet when the bulk transfer is finished.

BVS protocol [101] was intended for media streaming and also employs NACK to inform the sender that packet re-transmission is required. BVS applies retransmission only to prioritized packet(s). Furthermore, Fox [105] implements NACK as an option for providing a more efficient TCP operation over a network with a high bandwidth-delay product, such as satellite networks. TCP with the NACK option sends the NACK packet to the sender when the packet sequence is incomplete. Tezcan et al. [106] surveyed various acknowledgement-based protocols in wireless sensor networks. Work in [107] combines ACK and NACK to provide reliability in point-to-point data transfer within wireless sensor networks. NACK has also been used intensively in multicast networks.

The NORM (NACK Oriented Reliable Multicast) protocol [108] provides a reliable transport protocol from one or more senders to a group of receivers in multicast networks. The protocol employs both ACK and NACK. NACK is sent to request missing or repaired data. The NORM protocol employs NACK in two ways; spontaneously, when loss is detected; and scheduled, using a timer.

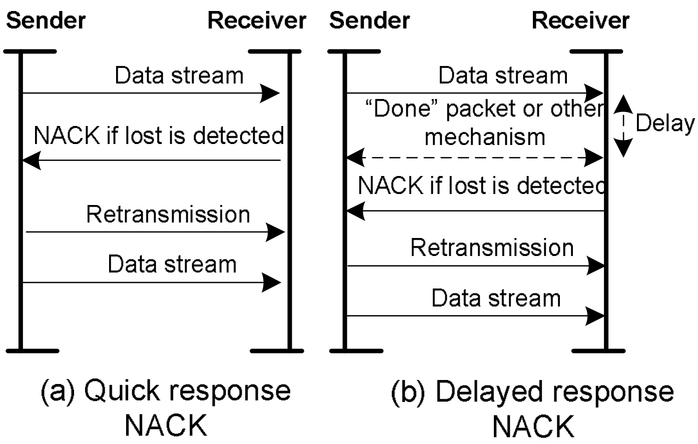


Figure 3.1: Existing NACK scheduling

NACK scheduling refers to the moment when a NACK packet is transmitted. NACK scheduling on existing protocol is classified in two types: quick response and delayed response, as shown in Figure 3.1. Quick response scheduling requires the receiver to send a NACK packet to the sender as soon as packet loss is detected. In contrast, the delayed response scheduling receiver sends a NACK packet at a particular time or to a specific event. The NACK scheduling in UDT [100], BVS [102] and TCP protocol in [105] are categorized as quick response. Meanwhile, NACK scheduling in RBUDP [101] is delayed response scheduling. NORM [108] implements both quick and delayed response NACK.

3.5 WiMAX Cross-layer techniques

Many researchers explore the cross-layer techniques in WiMAX; examples of which are presented in literatures [19, 109-114]. Most WiMAX cross-layer approaches explore PHY or MAC layer properties. For instance, Noordin and Markarian [19] use PHY properties to support MAC layer scheduling. The proposed scheduler implemented a cross-layer optimizer between MAC and PHY layers in order to maximize WiMAX performance. The optimizer collects data from both layers and returns the optimized parameters for bandwidth allocation in MAC layer as well as the coding selection in PHY layer. Besides optimizing the schedulers [19], the cross-layer is also employed to enhance data transfer related to video codec [109-111]. Meddour et al. [109] used MAC properties to optimize unicast and multicast video streaming, while Martini and Hewage [110] explore the MAC layer burst to insert sync-words for video-frame synchronization enhancement. Al-Jobouri et al. [111] adjusted the rateless code by using FEC information that reflects the channel condition. The handover process can also take advantage of the cross-layer techniques to become seamless [112].

Besides the MAC-PHY cross-layer to enhance WiMAX scheduling, MAC-Network cross-layer provides a seamless handover and PHY-Application or MAC-Application cross-layer enhances video transfer. Moreover, the cross-layer approach can be used to enhance the performance of multi-type of networks. Castro and Fernandez [113] employed a cross-layer technique to dynamically optimize the hybrid satellite-WiMAX network's capacity. Layers 2 and 3 combinations in [114] improve the vertical handover performance between WiMAX and 3G networks.

The work detailed in Chapter 4 completes the cross-layer schemas by proposing the cross-layer between MAC and transport layers. The proposed Transport MAC cross-layer protocol provides high performance end-to-end transport layer connection in WiMAX networks to replace the existing UDP protocol. The protocol does not aim to compete with the existing cross-layer works, as each has a different emphasis. The MAC-PHY, MAC-Application and the proposed cross-layer could be incorporated to achieve the anticipated performances.

The cross-layer between the MAC and transport layer protocol has been used explicitly in some existing reliable transport layer protocols that employ congestion controls, such as in [115, 116]. Ye et al. [115] used the cross-layer method to provide fairness for some TCP flows. Work in [116] proposed WCCP (Wireless Congestion Control Protocol), which is effective only for a static ad hoc network and adjusts the sending rate based on channel utilization. However, reliable-based protocols are not suitable for multi-sources real-time video transmission over WiMAX, as those protocols exert tremendous delays. Unlike reliable protocols, the proposed cross-layer does not explore channel quality to support congestion control. The protocol is intended to improve the existing unreliable protocol; therefore, the implemented methods should not remove the nature of the unreliable protocol. The congestion avoidance is performed as simply as possible and the retransmission effort is performed only once.

3.6 Error concealment techniques

Although multiple layer improvements are implemented, lost packets may always exist in wireless video transmission. Error concealment is considered as the last means to overcome packet loss that reduces video image quality. Moreover, it replaces the lost packet by using information from others. There are two well-known error concealment techniques: the frame copy method and motion vector copy method. Both have been implemented in the H.264/AVC codec [117, 118].

Most researches focus on motion vector based error concealments, whether spatial error concealment, temporal error concealment or their modification [117-125]. Chien et al. [117] assumed that a single video frame coded to a single packet would maximize the bandwidth. The authors proposed an algorithm to select the maximum motion vector different and refine the lost area recursively. In [118], multi-frame error concealment was proposed. The proposed algorithm estimates the distortion of both the lost frame and its succeeding frame, and selects the recovering modes. The author also introduced complexity adaptation to achieve optimal complexity-distortion. Pyun [119] proposed an error concealment aware error resilient streaming video system. The streamer selects the suitable error concealment method per macro block and sends the selected code to decoder. Suissa et al. [120] proposed a full search and a spiral search to conceal the lost area using a block motion copy.

Conversely, the frame copy method is considered too simple, since it is uses part of another image to replace the lost one; thereby producing a rough contour. Many papers use the frame copy method as the lowest performance to compare and only few research use this method. For example, Kim, Ryu and Jayant [126] used this method within frame smart skipping to conceal frames lost during the handoff process. Although the motion vector (MV) error concealment method is the more popular of the two, its implementation has drawbacks as its computational cost is higher [119]. Bo and Gharavi [124] reported that MV-based error concealment, for example optical flow, was not always preferable to frame copy as the larger motion picture will reduce the accuracy of MV. Chapter 7 explores post-decoding error concealment methods based on the partial frame copy method and applied after decoding process. The proposed error concealment can be used to complement the existing methods and the additional solution if the current error concealment techniques fail.

3.7 Summary

This chapter has described related works on WiMAX-based surveillance, WiMAX bandwidth request and scheduling, unreliable protocol improvements and error concealment techniques.

Research on WiMAX-based surveillance remains limited. The existing works assess either WiMAX implementation or the received video quality enhancements.

WiMAX bandwidth request mechanisms and scheduling algorithms have been presented. The bandwidth request mechanism in WiMAX standard comprises contention request, unicast request and piggybacking. Most existing works on bandwidth request require major changes to the standard, such as the replacement of TBEB algorithm. On the other hand, the WiMAX scheduler is left empty as it is implementation-dependent. Existing schedulers comprise legacy schedulers, hybrid schedulers, cross-layer-based schedulers and packet-based schedulers.

There have been many researches on improving unreliable transport layer protocols, either using TCP properties, error correction within transport layer packets or applying retransmission. However, there was a limited study on transport layer protocol designed specifically for WiMAX-based video transmission.

Error concealment can be used as the last means to recover packet loss, and employing such techniques can improve video surveillance.

Chapter 4
MAC layer: Bandwidth allocation

4.1 Introduction

WiMAX devices have been used widely in the market [127]. WiMAX-based video surveillance products have also been available, as in [128, 129]. The acceptance of WiMAX in the market, as well as the availability of WiMAX products, contributes to the possibility of adopting it for surveillance application. However, since WiMAX is designed to accommodate various applications with different quality of service (QoS) requirements [12], dedicated surveillance network implementation of WiMAX may not achieve optimum performance, as all SSs generate the same QoS requirements. Each SS transmits real-time video, which is classified as rtPS. There are no UGS, ertPS, or BE traffics. The scheduler cannot implement traffic type priority; therefore, service classification does not work as expected. Furthermore, since rtPS uses the unicast bandwidth request mechanism, all nodes will make unicast reservation when requesting bandwidth. This will generate a higher delay as the base station (BS) will poll all subscriber stations (SSs). Moreover, if the number of SSs is greater than the maximum polling capacity, then, besides leading to high delay, one or more SS(s) will experience bandwidth starvation.

Furthermore, when all nodes generate video traffic, the scheduler cannot differentiate traffic based on the standard class of service. The BS should take frame classification as the basis for priority selection.

This chapter addresses the issues mentioned above. The major contributions of this chapter are simplified bandwidth allocation architecture, packet-aware bandwidth request mechanism and packet-aware scheduling algorithms for dedicated video surveillance application with real-time uniform video traffics. The proposed architecture considers the uniformity of the traffic sources. Instead of using the rtPS class, this thesis implements best-effort services for all traffic sources. The proposed bandwidth request mechanism and the scheduling algorithms consider packets of the frame types. The proposed methods consider priority distinction for different frame types. The advantages of the proposed bandwidth allocation are transforming the general purpose WiMAX network into a special purpose dedicated video surveillance network. The proposed methods are designed to be applied to existing devices in the market, without the necessity of changing the hardware.

The remainder of this chapter is organized as follows: Section 4.2 outlines the details of the proposed WiMAX bandwidth allocation architecture, bandwidth request mechanism and scheduling algorithms; Section 4.3 explains the evaluation method; Section 4.4 presents the results of the method's evaluations; and Section 4.5 concludes the chapter.

4.2 Proposed methods

The proposed methods are intended for a dedicated surveillance network, which is assumed to be operational in a non-saturated condition. The SSs are connected to BS in point-to-multipoint topology. Each SS generates a video stream and transmits in the UL channel. Since there is only a single traffic source in each SS, the SS scheduler is not considered. BS receives video traffic from SSs; therefore, the traffic is assumed to have a uniform QoS requirement.

The proposed methods aim to minimize the transmission delay and maximize the quality of the received video. The objective of the simplified architecture is to maximize resources in BS, deactivate unnecessary services and minimize the processing delay. The proposed bandwidth request mechanism is intended to reduce transmission delay caused by the transmission request process. The proposed scheduler maximizes the quality of the received video. Since each SS produces a single video traffic, the grant allocation is based on GPSS.

In the following discussion, the bandwidth allocation architecture is initially defined, then suitable bandwidth request mechanism and scheduling algorithms are proposed.

4.2.1 Flat service class architecture

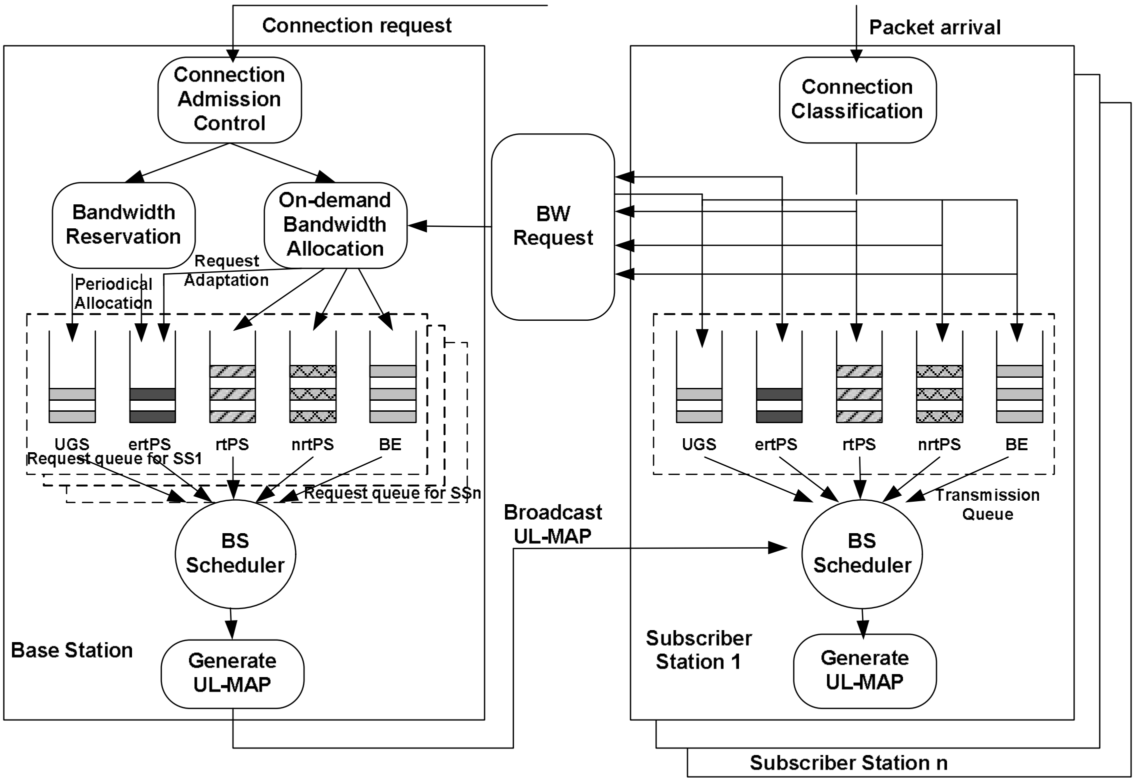
The assessed WiMAX-based dedicated surveillance network is assumed to use the same bit rate setting in all SS cameras. The generated traffic is uniform and behaves similarly. In such conditions, fairness is achieved when the class of service for all traffics is similar. Real-time video traffic is supposed to be classified as rtPS in the WiMAX standard, which results in the rtPS service being maximized and other services not being utilized. Since rtPS uses unicast request mechanism, BS should poll individually the SSs. The rtPS service also performs additional tasks to determine transmission parameters. These facts potentially result in WiMAX-based video surveillance with rtPS experiencing high delay.

BE outperforms rtPS for uniform video traffic. This may be caused by the WiMAX UL-sub-frame providing K-slots for contention requests and the BE service allowing SS to contend a request whenever there are data to send. Therefore, BE is used for all traffic sources. Moreover, BE is relatively simple to supply as it does not involve QoS negotiation and enforcement of traffic parameters [2].

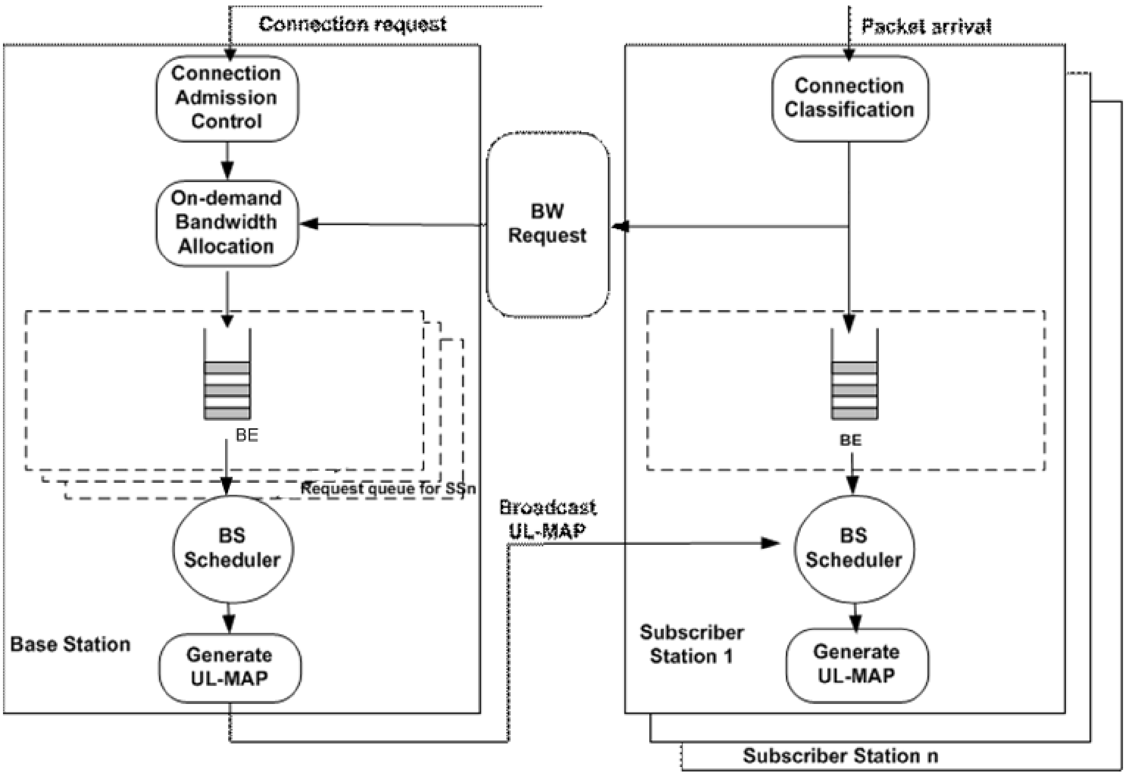
The bandwidth allocation architecture in WIMAX is redefined to retain only the BE service. Figure 4.1a shows the original architecture in WiMAX standard where various classes of service feed the schedulers. This standard architecture shows that scheduling occurs in both BS and SS sides. Each service type is bound to the respective bandwidth request mechanism. Since the UGS, ertPS, nrtPS and BE do not have traffic in the examined dedicated surveillance network, the allocated processor power to poll and memory resources for queue in BS and SS schedulers are wasted. Besides consuming BS and SS resources, the systems may exert more processing delays.

The redefined architecture avoids such waste by deactivating the un-used resources and allocating resources to the active service. Instead of providing various services for each SS, BS serves only BE service. BS employs only contention-based bandwidth request mechanism. Memory resources in BS are allocated to a single BE queue for each SS and the processor does not need to poll other services. The BS scheduler is designed only to serve BE. On the SS side, only the BE service is activated. Since there is only one traffic source in each SS, SS does not require a specific scheduling algorithm. Figure 4.1b shows the proposed architecture.

By implementing the proposed architecture, BS and SSs in the proposed dedicated WiMAX-based surveillance network are expected to provide more processing power and memory resources in order to minimize the processing delay.



(a). Original architecture, taken with permission from [17]



(b). Simplified architecture

Figure 4.1: Bandwidth allocation architectures

4.2.2 Packet-aware bandwidth request

The proposed packet-aware bandwidth request mechanism aims to reduce the delay. It implements two techniques, the reduced contention window to serve traffic from the prioritized frames (I-frames) and next-frame piggybacking to serve the non-prioritized video frames (P-frames and/or B-frames). The bandwidth request selection can be expressed as follows:

$$BW\ request = \begin{cases} Reduced\ contention\ window, if\ frame\ type\ = I \\ Next\ frame\ piggyback, If\ frame\ type\ = P\ or\ B \end{cases} \quad (4.1)$$

The following description explains the details of each mechanism.

a. The reduced contention window

As mentioned in Section 3.2, there are various techniques proposed to replace the existing truncated binary exponential backoff (TBEB) technique in the WiMAX standard. However, this thesis employs the existing TBEB in the standard, so that the proposed change is more applicable to the existing devices.

The TBEB algorithm determines the random integral number of contending nodes chosen from interval [0, W_i-1]. There will be a significant reduction of contending nodes, since only I-frames require contention bandwidth request. The faster the request is sent, the lower the bandwidth request delay. Therefore, the CW size reduction is proposed. Unlike the method proposed in [12], which adjusts dynamically the timeout, a fixed CW size adjustment is chosen to avoid unnecessary delays, as the timeout adjustment is restricted by WiMAX-frame allocation.

The new backoff window size can be assigned as a reduction percentage of P and B-frames: $(\sum P\ size + \sum B\ size)/I\ size$ within one group of picture (GOP). The CW reduction is proposed as the number of contending SSs decreases. However, in order to reduce calculation delay, a 50% reduction is used. Therefore, the new random integral number for contending nodes is chosen from the interval $[0, 0.5(W_i - 1)]$. If collision occurs, the increment will be $W_i' = 2^{i-1}W_{i-1}', i \neq 0$, where $W_i' = 0.5\ W_i$.

By using the TBEB analysis proposed by Chen and Tseng [13], the probability of a successful request, P_s is a function of the number of available request slots s and the expected number of contending nodes, n_{exp}. That is,

$$P_s = \left(\frac{s-1}{s}\right)^{n_{\exp -1}} \tag{4.2}$$

where the expected number of contending nodes is defined as

$$n_{exp} = n\frac{n_{tx}}{n_{tf}}\left(1 - e^{-\lambda . n_{tf} f}\right) \tag{4.3}$$

Here, λ is distributed Poisson arrival rate, f is the frame duration, n is the number of SS, and n_{tx}/n_{tf} is the probability that one SS sends a bandwidth request. The average number of requests n_{tx}, and the average request processing time n_{tf}, are functions of the contention window $W_i, i = 1,2 \dots, m$, and collision probability, c.

$$n_{tx} = \sum_{i=1}^{m} i(1-c)c^{i-1} + (m+1)c^m \tag{4.4}$$

$$n_{tf} = \sum_{i=1}^{m}(1-c)c^{i-1}\left(\sum_{j=1}^{i}\frac{1}{W_j}\left(\sum_{k=1}^{W_j} k\right)\right) + c^m\left(\sum_{i=1}^{m+1}\frac{1}{W_i}\left(\sum_{j=1}^{W_i} j\right)\right) \tag{4.5}$$

The TBEB performance for the arrival rate and CW reduction is depicted in Figure 4.2. If initial arrival rate and contention window are λ=120 and W_o=8 respectively, the replacement of P-frames bandwidth request using next frame piggyback reduces the arrival rate. The TBEBs for reduced arrival rates (λ=100 and λ=80) produce higher successful probability than TBEB with λ=120. Arrival rate and CW reduction from 8 to 4 provide more improvement in the successful probability.

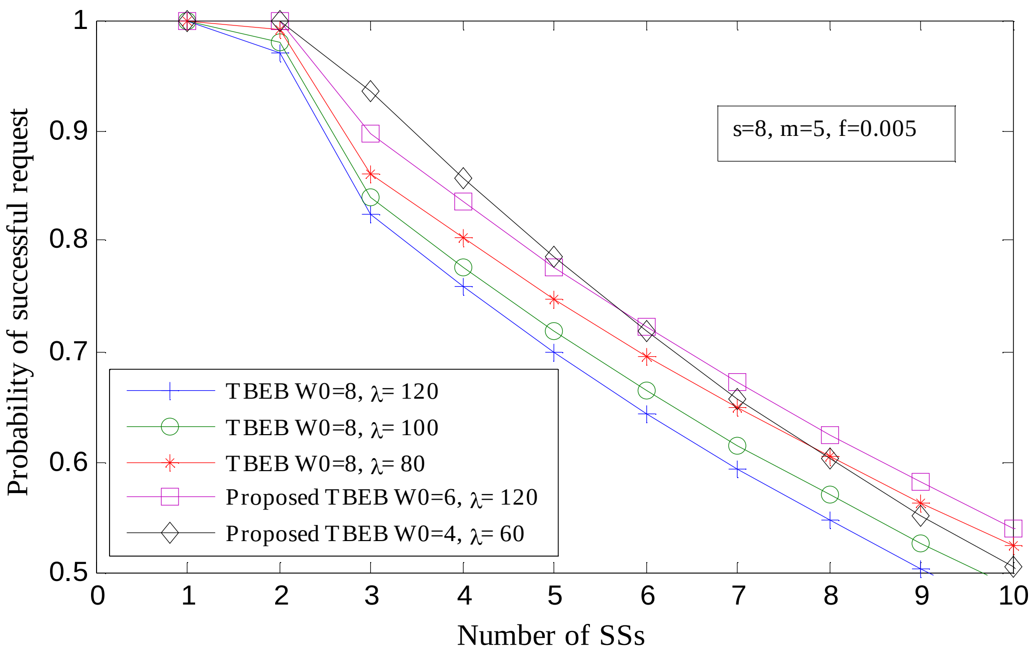


Figure 4.2: Performance of TBEB

Although the proposed TBEB performance degrades when the number of SSs increases, the surveillance network is assumed working on a non-saturated network with an acceptable maximum number of SSs.

b. Next-frame piggyback

This thesis proposes the next-frame piggyback method to serve traffic from the P-frames. The method allocates the bandwidth request for the next frame $n + 1$ in burst of frame n as discussed in [77]. However, the capacity of the next frame $n + 1$ is predicted, as the size of frame $n + 1$ is unknown when sending frame n.

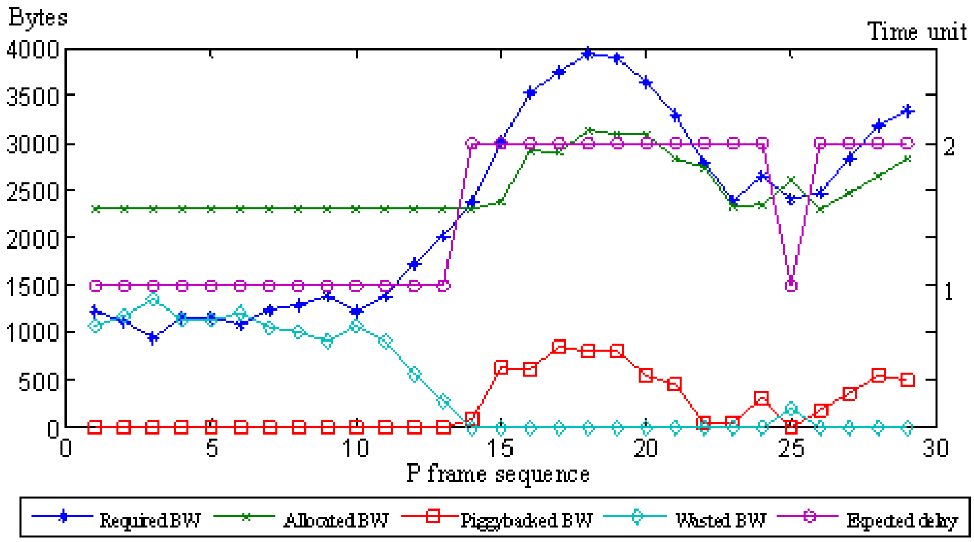
If the number of bytes in the current queue is B_n, the allocated bandwidth is B'_n and the predicted next frame bytes is B'_{n+1} then the number of bytes of the piggybacked request is:

$$PB_n = (B_n - B'_n) + B'_{n+1} \tag{4.6}$$

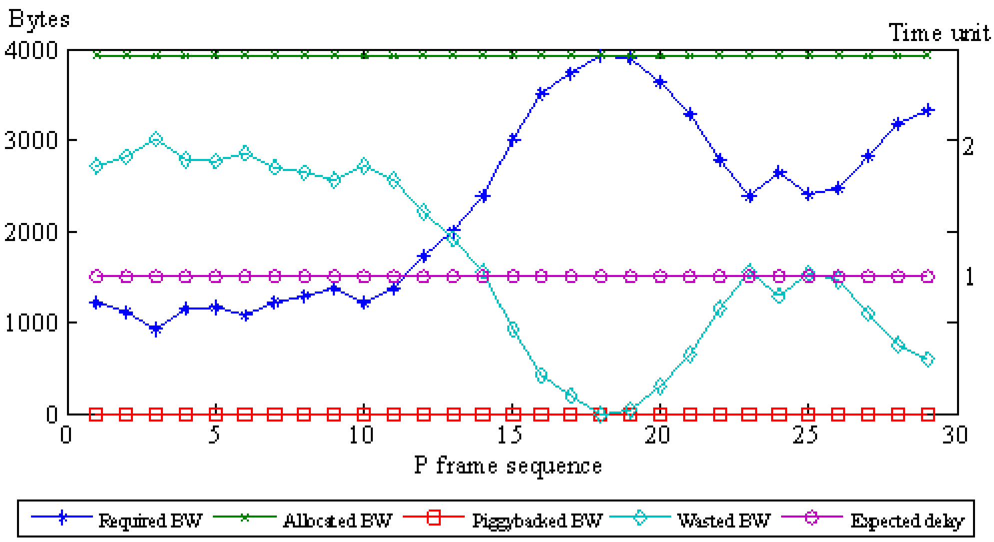
Since the value of B'_{n+1} is predicted, there is a chance that the predicted bytes are less or more than the actual ones. If Equation 4.6 for the next frame request is rewritten to $PB_{n+1} = (B_{n+1} - B'_{n+1}) + B'_{n+2}$, the first right part determines whether the allocated bandwidth satisfies the SS need. If $B_{n+1} - B'_{n+1} = 0$, then the allocated bandwidth is precisely as required by SS. However, if $B_{n+1} - B'_{n+1} > 0$, the allocated bandwidth is less than bytes in the SS queue. Data will be sent in more than one burst, which potentially generates higher delay and jitter. However, if $B_{n+1} - B'_{n+1} < 0$ then the allocated bandwidth is higher than the available data. Consequently, $(B'_{n+1} - B_{n+1})$ bandwidth is wasted.

Figure 4.3 presents the results of the constant B'_{n+1} prediction using the average frame size and the maximum frame size for the first 29 P-frames of the video trace Akiyo [52] with GOP 30. The piggybacked value is based on the predicted value and data size in buffer. If the allocated bandwidth is greater than that required, the frame will be sent in one data burst and the anticipated delay is minimal. However, because the allocated bandwidth is higher than required, bandwidth is wasted. Conversely, if the allocated bandwidth is lower than the required one, piggybacking occurs and the frame will be sent in more than one burst, thereby increasing delay.

The average frame size prediction reduces wasted bandwidth, but increases delay (Figure 4.3a). The maximum frame size prediction minimizes delay, but wastes more bandwidth (Figure 4.3b).



a. Average value prediction



b. Maximum value prediction

Figure 4.3: Piggybacked and wasted bandwidth

4.2.3 Packet-aware non-sorting schedulers

As shown in Figure 4.1b, the scheduler inputs in the BS are best-effort traffic. BS is expected to process bandwidth requests as soon as they arrive and allocates the decision to the nearest DL burst. In order to achieve objective, the BS scheduler should have a low processing delay.

Most popular schedulers mentioned in Section 3.3, such as EDF and WFQ, are classified as sorting schedulers, where the bandwidth requesting nodes are populated and sorted according to a specific parameter. Basic schedulers, such as RR and FIFO, are classified as non-sorting schedulers.

According to Puschner [141], the average time complexity of sorting algorithms, such as bubble sort, insertion sort and selection sort, is quadratic in the number of elements. Algorithms with complexity $O(\log_N N)$ grow faster than linear ones, while distributed counting and radix sort are almost linear in terms of the number of elements; for example, selection sort requires 1.40, 4.81, and 26.6 time units for sorting 5, 10, and 25 elements, respectively. The time unit is measured as a relative value to the lowest time (insertion sort with 5 elements corresponds to one time unit). The sorting schedulers populate all connections from nodes that have bandwidth request. Then, the schedulers sort the list of connections based on particular parameter. The sorting process potentially generates processing delay in BS and postpone bandwidth grant in the nearest DL-MAP, which, in turn, increases transmission delay.

By considering the required time for sorting and the fact that all traffic sources have similar requirements, the $O(1)$ non-sortingschedulers may perform better than sorting schedulers. The proposed schedulers consider processing time and transmission delay for high-capacity requests, such as I-frames, which have higher capacity than P or B-frames. Allocating such a large amount of bytes to insufficient bandwidth can cause a frame to be transported in different bursts. Sending one frame using separated bursts may result in high delay and loss; therefore, the proposed schedulers prioritize important frames and avoid sorting. The schedulers are referred to as packet-aware non-sorting schedulers.

a. RR-based scheduler

RR scheduler has disadvantages, as BS must check whether all the registered nodes in scheduling table have a bandwidth request. The more registered nodes, the more time is spent checking. However, RR do not need to populate the requesting nodes, it only checks the node sequentially. Therefore, the required checking time may be lower that needed to populate and sort. The sorting schedulers have higher delay than RR for uniform traffics, especially when the number of SSs is small.

The RR-based packet-aware non-sorting scheduler works as follows. BS checks whether a node has made a bandwidth request. If the request exists and the bandwidth is for a prioritized frame, a bandwidth allocation decision is made directly. However, if the bandwidth request is for non-prioritized frame, the request will be suspended temporarily. If all nodes have been checked, the suspended requests are then processed and bandwidth requests are allocated. The proposed algorithm is shown in Figure 4.4. One of the advantages of the proposed RR-based scheduler is that bandwidth for prioritized packet is directly allocated.

Proposed RR based packet aware non-sorting scheduler

```
begin
    for i=0 to n-1 do
        begin
            if (node[i].Bwrequest>0 && frame=='I') then
                UL-MAP←Allocate(node[i].Bwrequest.length);
            else if (node[i].Bwrequest>0 && frame!='I') then
                BWarray←Allocate(node[i].Bwrequest.length);
        end;
        if(i==n-1 && Bwarray.length>0) then
            for j=0 to Bwarray.length-1 do
                begin
                    UL-MAP←Allocate(BWarray[i].length);
                end;
    return UL-MAP;
end;
```

Figure 4.4: RR based packet-aware non-sorting scheduler

b. FIFO-based scheduler

FIFO scheduler processes bandwidth requests based on the earliest arrival time. BS should decide how long to wait for a new request before deciding to grant the bandwidth and sending the burst. If BS fills the burst until all bandwidth is consumed, it will produce high delays when the request is not frequent. Therefore, BS should limit the waiting time by checking the next SS when there is no incoming request. In the proposed scheduler, for each request received by BS, if the requested bandwidth is for the prioritized frames, a bandwidth allocation decision is made directly. But if it is from non-prioritized frames, the request will be suspended. If there is no subsequent request from the last node served, the scheduler checks the next node. If the total number of requests and the checked SSs is equal to the number of registered SSs, the suspended requests are processed and bandwidth requests allocated. The proposed FIFO-based packet-aware non-sorting scheduler is shown in Figure 4.5.

Proposed FIFO based packet aware non-sorting scheduler

```
begin
    for i=0 to n-1 do
        begin
            if (Request.exist=='true') then
                node[i]=Request.first.node();
                Request.removefirst;
            if (node[i].Bwrequest>0 && frame=='I') then
                UL-MAP←Allocate(node[i].Bwrequest.length);
            else if (node[i].Bwrequest>0 && frame!='I') then
                BWarray←Allocate(node[i].Bwrequest.length);
        end;
        if(i==n-1 && Bwarray.length>0) then
            for j=0 to Bwarray.length-1 do
                begin
                    UL-MAP←Allocate(BWarray[i].length);
                end;
    return UL-MAP;
end;
```

Figure 4.5: FIFO-based scheduler algorithm

4.3 Performance evaluation

In order to evaluate the proposed methods, the NS-2 simulations using the scenario outlined in Section 2.11 are performed. First, the performances of the proposed flat BE service architecture were evaluated and compared to the standard architecture with rtPS for uniform surveillance traffic. BE uses a contention request, while rtPS uses a unicast request. Afterwards, various bandwidth request mechanisms, including the proposed one, are examined using previous architecture. Finally, the proposed schedulers were examined and compared to RR, FIFO, EDF and frame-based schedulers using both the contention request and the proposed bandwidth request mechanism.

The other sorting schedulers are represented by EDF and frame-based schedulers, while weighted schedulers, such as DRR, WRR, cross layer scheduler and WFQ, are not suitable for uniform traffic.

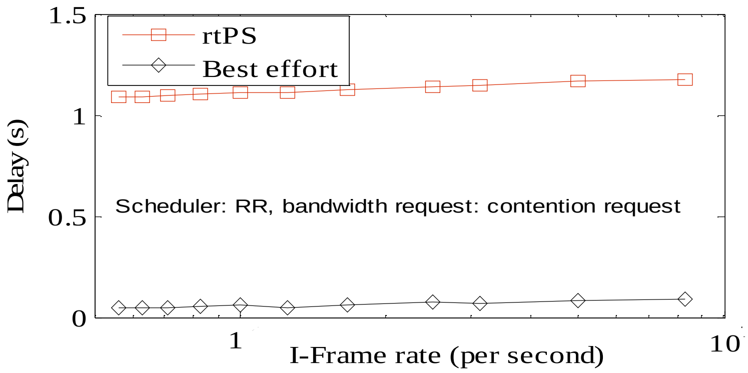
4.4 Result and analysis

4.4.1 Performance of the flat BE service architecture

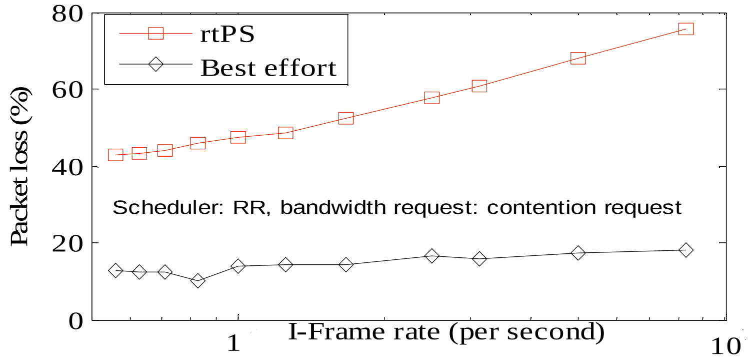
Figure 4.6 demonstrates the performance comparison between the proposed BE service architecture and the standard rtPS service in terms of delay and packet loss. Since RtPS service is designed for video traffic to have greater priority than non-rtPS traffic, the service is only suitable for non-uniform traffics. When the evaluated traffic has similar requirements, rtPS requires BS to poll all SSs; consequently, SS experiences high delay.

In contrast, BE service allows SS to contend any time to send data. Since the BS allocates K-slots of its UL-sub-frame for contention, the opportunity to send data successfully is greater than waiting to be polled. The polling system should catch the strict timing of the UL-sub-frame. If polling misses the closest DL-MAP, then SS loose the grant and delay increases (Figure 4.6a). Consequently, data in the SS buffer could be higher than the next allocated bandwidth. As a result, data could be sent in more than one burst or it may overload the SS buffer, leading potentially to high packet loss. Figure 4.6b shows that simulated rtPS service for uniform video traffic may generate high packet loss.

Conversely, BE generates lower delay and packet loss than RtPS. The obvious reason for this is that the network operates under non-saturated conditions with a small number of SSs; thus, the probability of a successful contention request is high. Each SS receives bandwidth grant faster than served by rtPS. On average, BE service experiences 1.06s lower delay and 38% lower packet loss than RtPS. Moreover, the result in [62] confirms that the contention request outperforms the unicast request for a non-congested network. Based on this result, the next experiments use BE service.



a. Packet delay



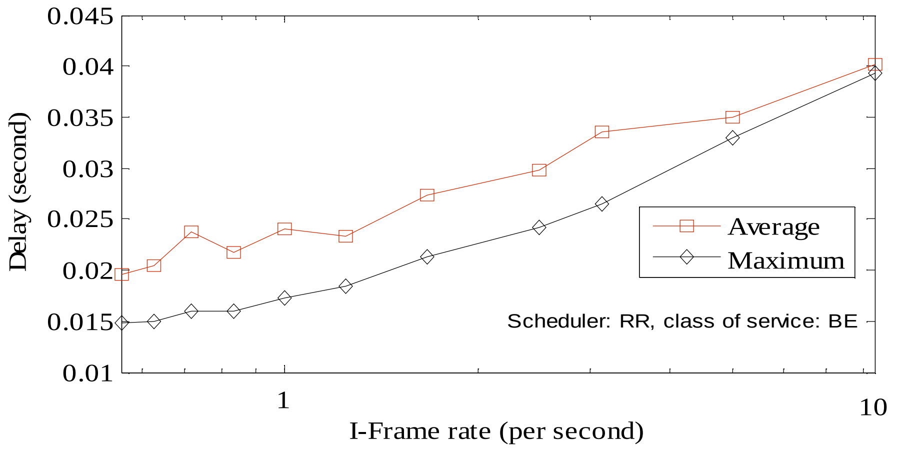
b. Packet loss

Figure 4.6: Performance of BE vs rtPS

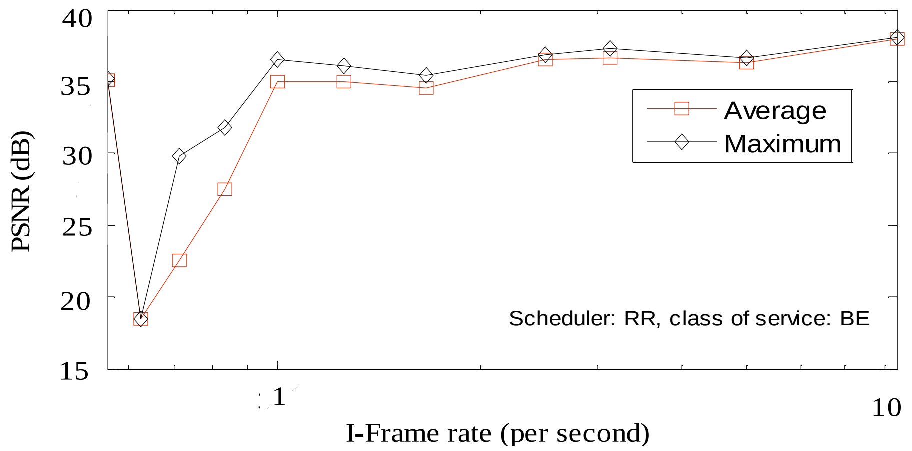
4.4.2 Analysis of bandwidth request mechanisms

The performance of the proposed bandwidth request mechanism for P-frames is affected by the piggybacked byte prediction. Figure 4.7 shows that the maximum P-frame size prediction exerts lower delay than that using the average values. Section 4.2.2 highlights that sufficient bandwidth allocation produces regular UL burst and avoids separated frame transmission. The greater the bandwidth allocated, the more opportunities there are to send data. Consequently, sufficient bandwidth allocation reduces packet loss and improves video quality. Since maximum P-frame size is not significant, compared with I-frame size, the bandwidth waste in maximum value prediction is acceptable.

Figure 4.7b presents the average PSNR values of the received video for both average and maximum predictions of next-frame piggybacking. Maximum prediction results are better PSNR than the average prediction. Although PSNR experiences irregularity when I-frame rates are lower than 1, this may be caused by the received video suffers from high non-decodable frames, which makes PSNR values drop to about 18dB. The occurrences were repeated in other experiments, which show that GOP values from 30 to 40 (I-frame rates 0.833 to 0.625) are sensitive to packet loss.



a. Packet delay

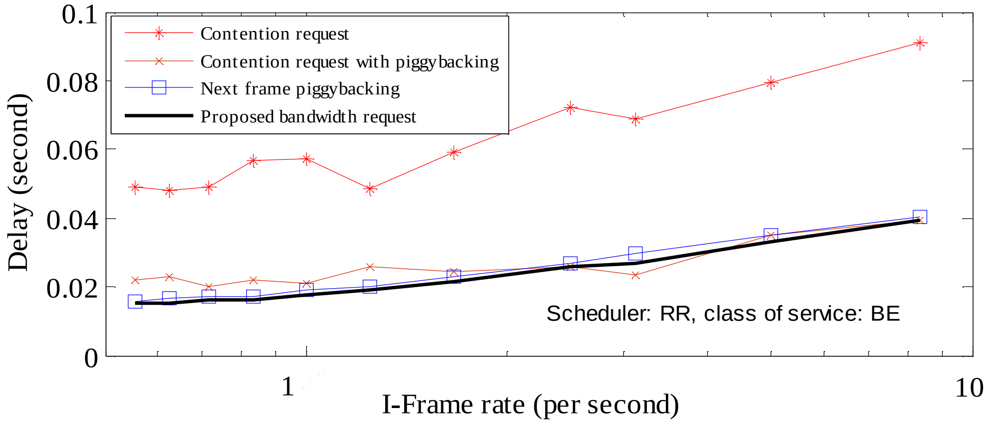


b. PSNR

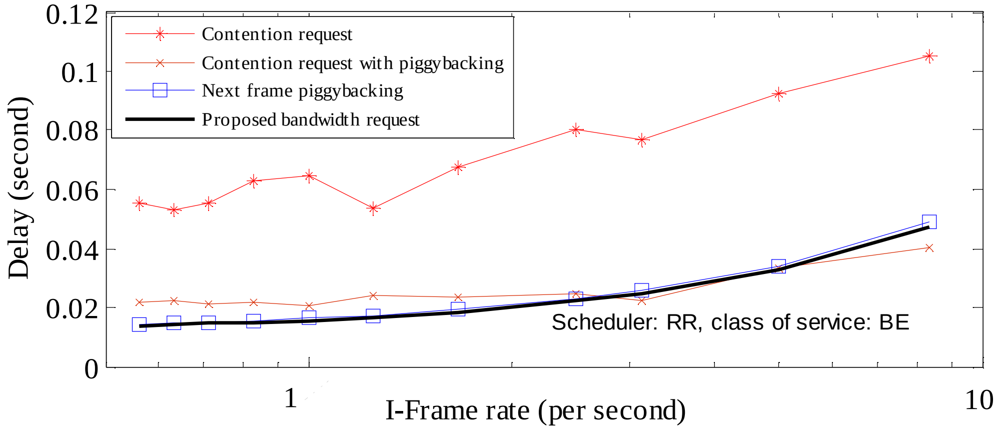
Figure 4.7: Performance of the predicted next frame piggybacking

Figure 4.8 presents the delay performances of unicast polling and contention request [12], contention request with piggybacking [66], next-frame piggybacking [77] and the proposed bandwidth request mechanism with a constant predicted value of 3500 bytes. This value is chosen from the average and maximum P-frame sizes.

Since the proposed bandwidth request was intended to reduce delay, as mentioned in Section 4.2, the proposed mechanisms were compared based on packet delay and frame delay performances.



a. Packet delay



b. Frame delay

Figure 4.8: Performance of bandwidth request mechanisms

The proposed bandwidth request achieves the lowest packet and frame delays for almost all I-frame rates. The main reason is that the proposed method has lower request contenders than contention request and piggybacking methods as only I-frames take the request opportunities. Consequently, the successful probability of the request is higher, which leads to fewer delays.

The next-frame piggybacking experiences slightly higher packet and frame delays than the proposed method because it serves I-frames and P-frames in the same way. This method piggybacked I-frames several times, as the frame sizes are larger than P-frame sizes.

The higher the I-frame rates, the greater the bandwidth requested by nodes. BS may allocate bandwidth less than the SSs requested as the available bandwidth is distributed among the SSs. As a result, delays increase in tandem with I-frame rates.

4.4.3 Analysis of Scheduling Algorithms

Since the proposed schedulers are packet-aware schedulers, they were compared to the frame-based scheduler [20]; a state of the art packet-aware scheduler. Although priority-based EDF [21] is also a packet-aware scheduler, it is similar to [20] for SSs with similar traffic requirements.

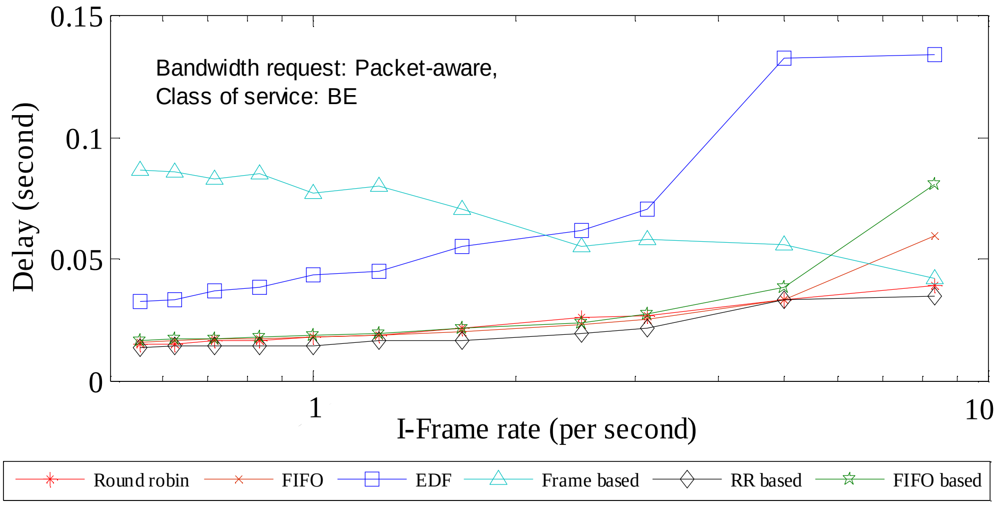
The proposed schedulers are non-sorting schedulers; therefore, the schedulers were also compared to EDF, as being representative of sorting schedulers. The scheduler performance assessment was conducted by using both the contention request mechanism and the proposed bandwidth request mechanism.

Table 4.1: Performance of scheduling algorithms using contention request

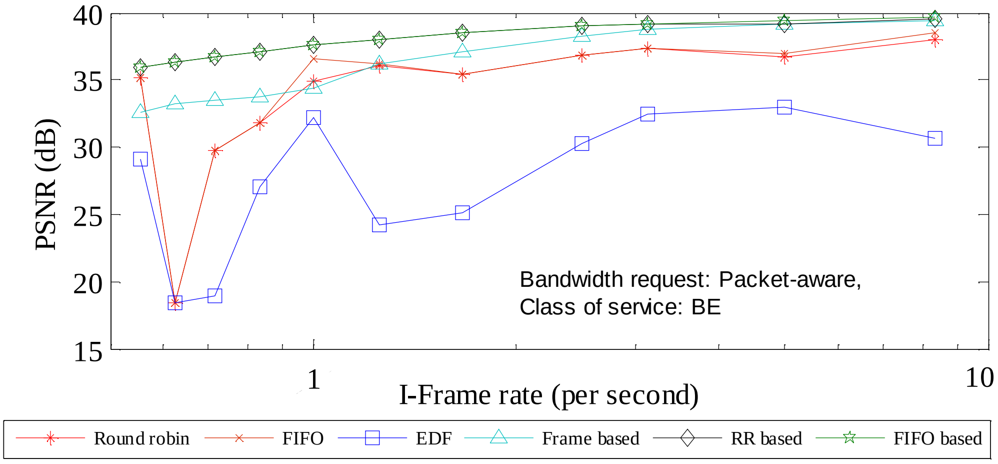
Parameter	RR	FIFO	EDF	Frame based	RR based	FIFO based
Delay (s)	0.062	0.067	0.065	0.063	0.063	0.065
PSNR (dB)	24.29	25.14	24.72	25.32	26.26	26.50

Table 4.1 depicts the average delay and PSNR performances of the schedulers using contention request mechanism. As stated previously, bandwidth request mechanism is aimed at reducing delay and the proposed schedulers are aimed to improve the video quality (PSNR). Therefore, this thesis focuses on PSNR achievement when assessing the proposed schedulers. Table 4.1 shows that the contention request assessment proves the excellent performance of the proposed scheduler; RR-based and FIFO-based.

Both proposed schedulers outperform existing schedulers in terms of video quality. The frame-based scheduler, who also prioritizes I-frames, achieves third place for video performance. It shows that packet-aware schedulers are suitable for WiMAX with SSs generating similar video characteristics. The proposed schedulers are able to achieve the highest PSNR because they allocate the entire prioritized packets in the first place. The frame-based scheduler does the same by setting a minimum deadline for I-frames. However, they do not achieve optimum performance as the deadline stamping in the sender and the sorting process in BS contribute to processing delay and affect the overall performance. The RR, FIFO and EDF experience low PSNR as those schedulers do not make sufficient effort to increase the received video quality.



a. Delay



b. PSNR

Figure 4.9: Performance of scheduling algorithms

Figure 4.9 depicts the scheduler performances using the proposed bandwidth request mechanism. The proposed scheduler outperforms the existing schedulers for both delay and PSNR. Although FIFO-based scheduler delay is greater than RR and FIFO for higher I-frame rates, its video performance is much better than both schedulers. Both proposed RR and FIFO-based schedulers have a similar video performance. Frame-based scheduler delay decreases if I-frame rate increases. Since it prioritizes I-frames with zero deadlines, the sorting process is much easier when traffic has more I-frames. Consequently, it produces slightly lower PSNR than the proposed schedulers.

In contrast, EDF experiences the highest delay and the worst PSNR. The deadline stamping does not work for uniform traffic because all traffic has similar deadline requirements. The EDF sorting results in P-frames having more priority than I-frames. The possibility of I-frames being transported in separated bursts increases, which worsens latency and degrades PSNR.

The irregularity of PSNR values in Figure 4.9b, which drops to about 18dB,may be caused by consecutive packet losses. These losses make the received frame un-decodable and consequently, the average PSNR values dropped to the lowest figure.

4.5 Summary

This chapter proposed flat BE service architecture, bandwidth request mechanism and scheduling algorithms for WiMAX, which is used for the infrastructure of the real-time dedicated video surveillance.

The proposed flat best effort architecture deactivates the un-used services and allocates resources for the active service. The proposed architecture aims to maximize WiMAX resources so that the surveillance application run on the network performs at an optimal level.

Following the architecture, a packet-aware bandwidth allocation that behaves differently for different frame types was implemented. The proposed bandwidth allocation consists of a packet-aware bandwidth request mechanism, RR and FIFO-based packet-aware non-sorting schedulers. The packet-aware bandwidth request mechanism uses reduced CW for the prioritized frames and piggybacks the requested bandwidth for non-prioritized frames. Both RR and FIFO-based schedulers allocate bandwidth to prioritized packets before serving the non-prioritized ones.

The evaluation shows that the proposed techniques outperform the existing methods. The BE service is more suitable for WiMAX with SSs generating similar video traffic to the rtPS service. BE, with the proposed architecture, is able to improve significantly the overall performance. The combination of the reduced TBEB and the next-frame piggyback is also able to reduce significantly bandwidth request delay. Both RR and FIFO-based packet-aware schedulers are able to improve PSNR values of the received video, whether by using contention request or the proposed bandwidth request mechanism.

Maximum performance is achieved when the proposed methods of flat best effort architecture, packet-aware bandwidth request mechanism and packet-aware schedulers are working together.

Chapter 5
Transport layer: UDP improvement

5.1 Introduction

TCP provides high reliability data transfer, ensuring that each frame is received successfully and sequentially. However, TCP is not suitable for real-time video transmission, as wireless interferences and signal disruption may cause significant delay. UDP is the most common transport protocol for real-time video transmission over wireless networks [131]. However, UDP does not respond to network conditions, which can cause network congestion [132].

In order to gain maximum performance for the streaming application, transport protocol should be able to deliver video with sufficient quality while maintaining low delay connectivity. Many studies have been conducted to improve transport layer protocol performance, whether by employing error correction method, retransmission or congestion control services.

This chapter focuses on how to reduce packet loss by retransmitting dropped packets within one frame before sending the next frame. The MAC support is also discussed as the advance techniques to provide suitable bandwidth for the retransmitted packet(s). The remainder of this chapter is organized as follows: Section 5.2 describes the proposed retransmission techniques; Section 5.3 provides an advance technique that utilizes the MAC layer support to improve transport layer protocol performance; and Section 5.4 concludes the chapter.

5.2 Inter-frame retransmission protocol

The designed transport layer protocol is referred to as inter-frame retransmission protocol. The protocol employs negative acknowledgements: NACK, inter-frame NACK scheduling and congestion delay; the details of which are described in the remainder of this section.

5.2.1 Negative acknowledgement

The proposed protocol uses NACK to inform the sender that packet loss has occurred and the lost packet(s) should be retransmitted. The NACK packet contains either a list of indices of lost packets or the start and end indices of the lost packets. As soon as the sender receives a NACK packet, it resends the requested packets (Figure 5.1).

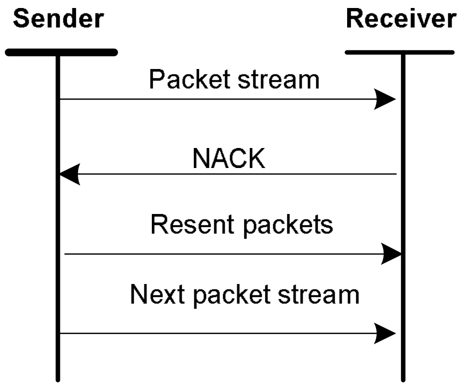


Figure 5.1: Negative acknowledgement

As discussed in Chapter 3, NACK is also implemented in RBUDP, UDT, BVS and NORM, which uses different NACK scheduling. This is discussed in the next section.

5.2.2 Retransmission scheduling

NACK scheduling refers to the moment a NACK packet is transmitted. NACK scheduling on existing protocol is classified into two types; quick response and delayed response, as shown in Figure 5.2.

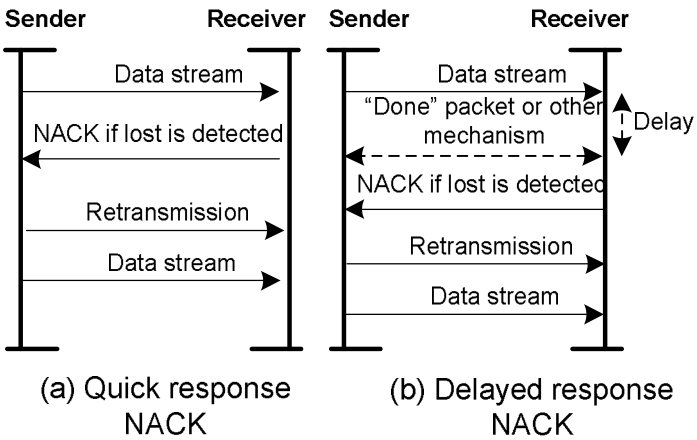


Figure 5.2: Existing NACK scheduling

Quick response scheduling requires the receiver to send a NACK packet to the sender as soon as it detects packet loss. In contrast, the delayed response scheduling receiver sends a NACK packet at a particular time or to a specific event. The NACK scheduling in UDT [100], BVS [102] and TCP protocol in [105] are categorized as quick response (QR). Meanwhile, NACK scheduling in RBUDP [101] is delayed response scheduling. NORM implements both quick and delayed response NACK. Since the NACK scheduling for media streaming, practically, it is only worth assessing quick response in existing protocols.

QR scheduling requires the receiver to send a NACK packet as soon as packet loss is detected. The packet loss information is determined by two values, the current and previous successfully received packet indices. The sender will check these values to decide which packet to retransmit. For instance, if the current packet index is 7 and the previous one is 4, then packets with indices 5 and 6 should be retransmitted. The advantages of QR are low NACK overheads and responding quickly to the loss. However, the receiver may generate more than one NACK packet for a frame, which requires more bandwidth and interrupts frequently the sender.

This section introduces an effective NACK scheduling for media streaming. The proposed scheduling is referred to as inter-frame (IR) scheduling. IR requires the receiver to populate packet loss indices within one video frame and sends a NACK packet at the end of a respective video frame reception. If packet loss does not exist in one video frame, then no NACK packet will be sent. A retransmission request using a NACK packet is applied only to prioritized packet(s). This thesis chooses I-frame packets as the prioritized packets. The advantage of the method is that a NACK packet will be sent only once for all lost packets within one video frame. By using this method, the sender will resend all lost packet at once. Figure 5.2 demonstrates how IR scheduling works.

Figure 5.3 shows the QR and IR NACK scheduling comparison for multi-packet losses. One video frame may be sent in several packets. The time distance between the last packet in one frame and the first packet of the next frame is called inter-frame gap (IFG). Figure 5.3a and 5.3b assume that the round trip time (RTT) is less than the IFG. Packets A and C within frame 1 are lost. In QR, NACK packets will be sent as soon as the receiver receives packets B and D. NACK packets may interrupt the sender frequently and may cause additional delay or another packet loss. Conversely, IR sends NACK and resends packets during the IFG, which increases the probability of using the channel when the sender is idle.

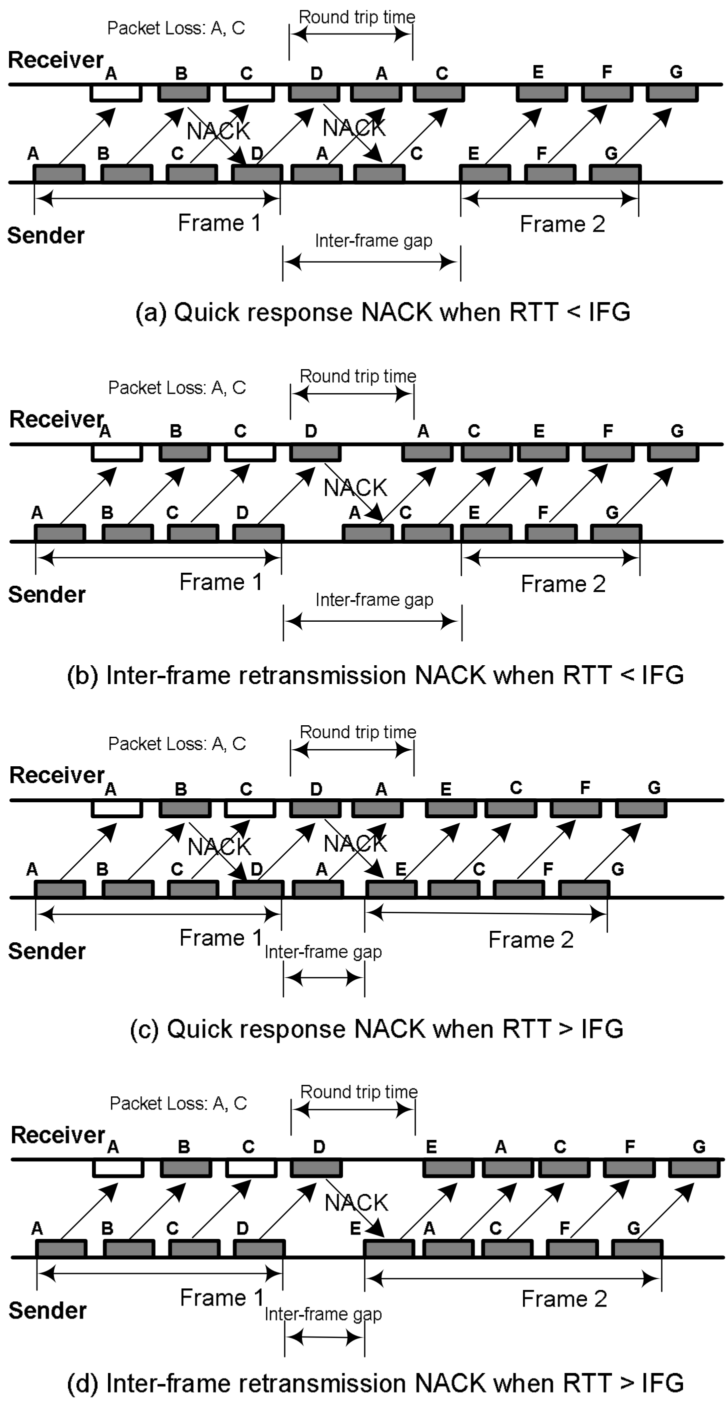


Figure 5.3: NACK scheduling

If RTT is greater than IFG (Figure 5.3c and 5.3d), IR interrupts the sender only once. Although IR seems to cause the next frame sending time to be longer, it is found that the sender processing time is more sensitive to NACK reception than to packet retransmission as retransmission routine is in the same location as transmission routine. IR has an additional requirement that the receiver should be able to detect the last packet in each frame. In case the last packet within a frame is lost, the lost packet will be retransmitted within the next frame.

5.2.3 Prioritized packets

Unlike TCP, which sends an acknowledgement for every received packet, the proposed protocol sends NACK packets only when packet loss occurs. However, if network congestion deteriorates, NACK packets may be sent more frequently as more loss appears. The frequent packet retransmissions may lead to high delay; therefore, in order to keep delays low, the NACK packet for a particular packet loss will be sent only once. The dropped retransmitted packet will be ignored.

Furthermore, the NACK packet reduction may be applied by sending NACK only for prioritized packets as video coding generates various frame significances. An additional packet header is required to flag whether or not a packet is prioritized. Simulation in this thesis uses MPEG4 video coding with IPP frame sequence. The prioritized packets are set as any packets corresponding to I-frames.

5.2.4 Congestion delay

Congestion delay (CD) aims to reduce the effect of sender interruption and avoid another packet loss by postponing the next packet transmission. CD also ensures that the current frame arrives before the next frame.

Figure 5.4a shows retransmission without CD. The sender sends packet E before retransmitting the lost packet C. Packet C, which belongs to the previous frame, may arrive after packet E, which belongs to the next frame, resulting in higher frame delay. In the worst case scenario, packet E can be lost during reception of a NACK packet. By using CD, packet C will be retransmitted before packet E, as highlighted in Figure 5.4b. This process results in lower delay on packet C and avoids the loss of packet E.

Although CD introduces more delay for packet E, a small CD value limits this additional delay.

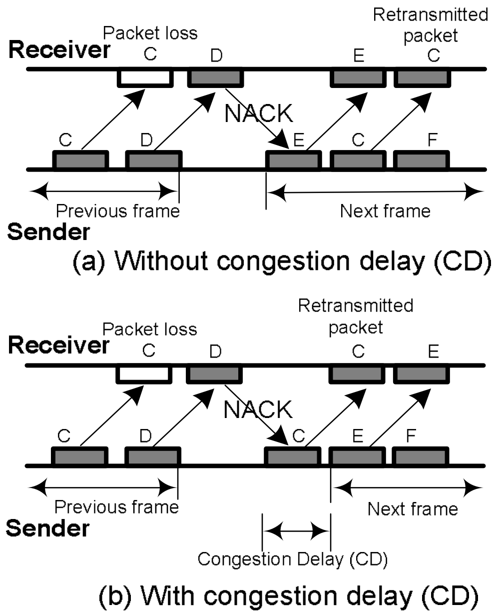


Figure 5.4: Inter-frame retransmission with CD

Moreover, CD acts as an instant congestion control mechanism by delaying next packet transmission in response to network congestion. CD produces temporary frame-rate reduction, as evidenced in Equation 5.1.

$$FR_{CD} = \frac{1}{{}^{1}/_{FR_{init}} + CD} \tag{5.1}$$

For example, if the initial frame rate (FR_{init}) is 25fps, and CD is 0.01s, then the frame rate caused by CD (FR_{CD}) is 20fps. This rate reduction gives the network time to reduce congestion, potentially reducing packet loss. The CD value should be smaller than the IFG in order to avoid the current frame competing with the next two.

5.2.5 Unicast CD for frequent retransmission

Transport layer protocol generates greater delays when video surveillance node uses higher bit rates video. There is more traffic flowing in the network, which produces longer queues at each network point. This situation makes NACK-based protocols perform worse as the retransmitted packets inject more congestion.

Unicast CD is able to compensate for the congestion increments. It aims to reduce network congestion in the current connection by decreasing transmission rate in another. Besides sending a NACK packet to the sender, the receiver can also send a unicast CD request to other predefined nodes. The requested node should decrease its rate by implementing CD. Consequently, additional traffic reduction from another connection will help to reduce network congestion.

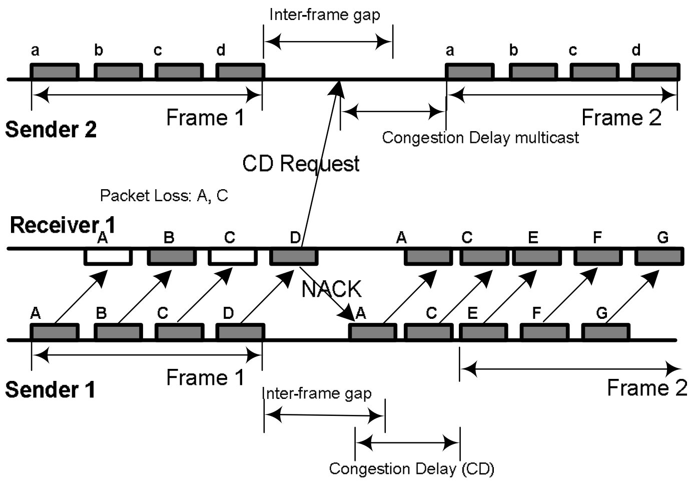


Figure 5.5: Unicast CD request

Figure 5.5 shows the unicast CD diagram. Receiver 1 experiences packet loss, then sends the CD request to sender 2 (unicast) and a NACK packet to sender 1. The CD request packet is similar to the NACK packet, with the exception that it does not contain list of the requested retransmitted packet. Sender 2 will delay its data transmission by applying CD. This delay gives traffic reduction and allows more bandwidth for sender 1. The greater the CD request, the more frequent the sender interruptions. Therefore, in the proposed protocol, unicast CD is optional.

5.2.6 Performance evaluation

The proposed protocol was evaluated using the NS-2 simulator in the scenario described in Section 2.11. It was then compared to UDP, TCP, BVS, DCCP and RBUDP. The evaluation used RR scheduler and contention request.

5.2.7 The protocol performance

The proposed retransmission was initially compared to quick response NACK for GOP 30. In addition to using delay and PSNR, the following parameters were also examined.

- Frame delay: the latest receiving time of the packets within one frame, subtracted by the frame time stamp
- Jitter: the absolute value of subsequent delay differences
- Fluidity: the frame distance obtained from the frame's received time subtracted from the previous frame's received time
- Packet loss: number of lost packets divided by total transmitted packets (in percentage)
- Cumulative throughput: total bits received during transmission.

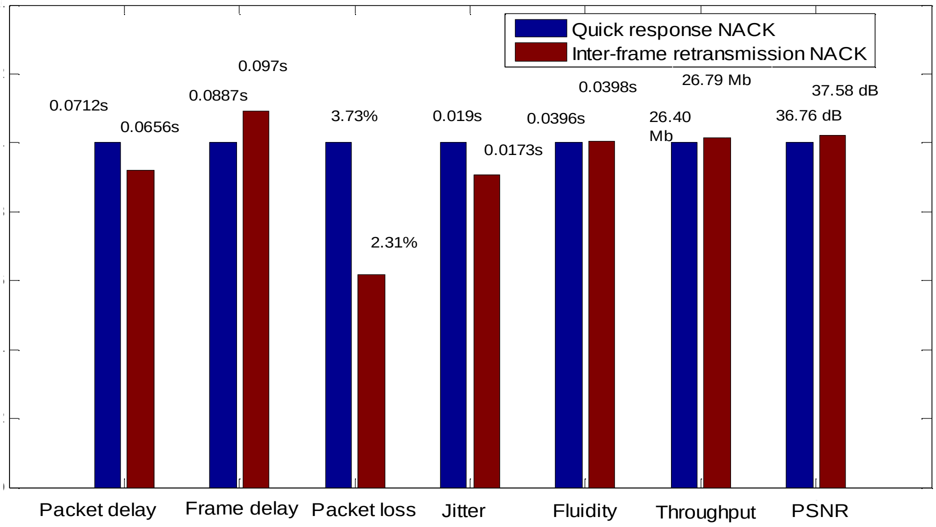


Figure 5.6: Comparison of QR and IR performance

Figure 5.6 presents the results. Inter-frame scheduling generates lower packet delay and jitter, less packet loss, closer fluidity to the original video, and higher cumulative throughput and PSNR than QR scheduling. Although its frame delay is slightly higher than QR scheduling, the overall performance of IR scheduling is better than QR.

By applying priority policy to IR scheduling (that is, sending NACK packets only if lost packets are parts of the prioritized frames), the protocol is able to reduce significantly packet and frame delays (about 10ms and 32ms on average).

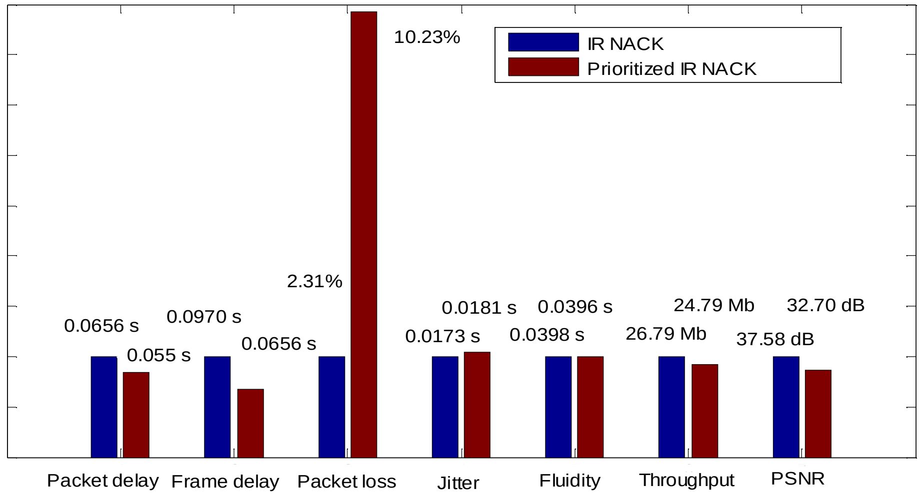


Figure 5.7: Comparison of IR and prioritized IR performance

Figure 5.7 shows the comparison of IR scheduling and prioritized IR scheduling. The prioritized one suffers higher packet loss, which reduces throughput and video quality. However, in real-time video transmission, low packet and frame delays are more important.

The delay parameters gained by prioritized scheduling as shown in Figure 5.7 should be suppressed further to produce better characteristics for video transmission purpose. CD is expected to achieve the anticipated performance. CD should be less than the frame distance, which means higher than 0 and lower than 0.004s (for frame rate 30fps). The smaller the value, the lower the effects on the next packet delay.

Various CD values for GOP 30 were tested, as detailed in Figure 5.8. The delay characteristics are relatively constant when CD values are fewer than 0.001s. However, they change alternately afterwards. On average, CD successfully reduces the delay of prioritized IR scheduling.

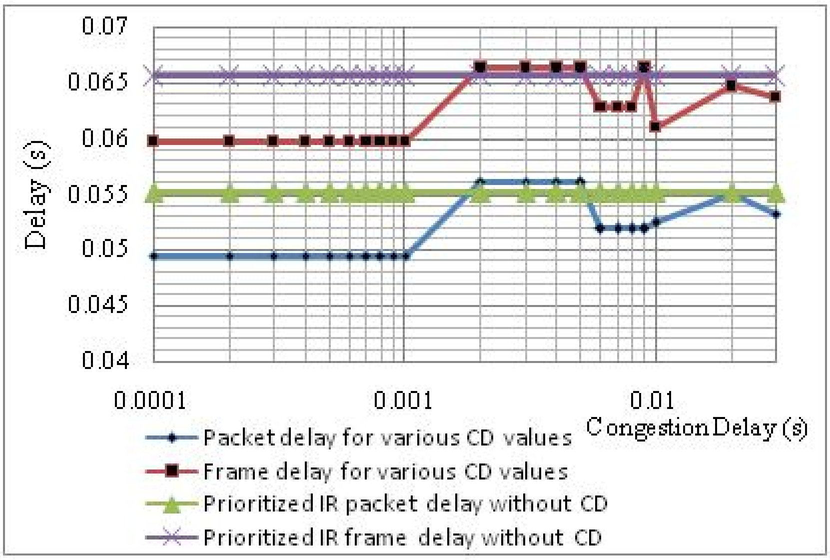


Figure 5.8: Congestion delay performances

Figure 5.9 depicts the performance enhancement of the proposed protocol by applying a 0.001s CD. The average packet and frame delays plunge respectively to 0.0495s and 0.0597s. The average jitter is also reduced to 0.0169s. Even if packet loss increased, causing a decrease in the cumulative throughput, the CD preserves prioritized frames better. This is shown by the increase of the PSNR, which means that the protocol successfully avoids more loss on prioritized frames and produces better video quality.

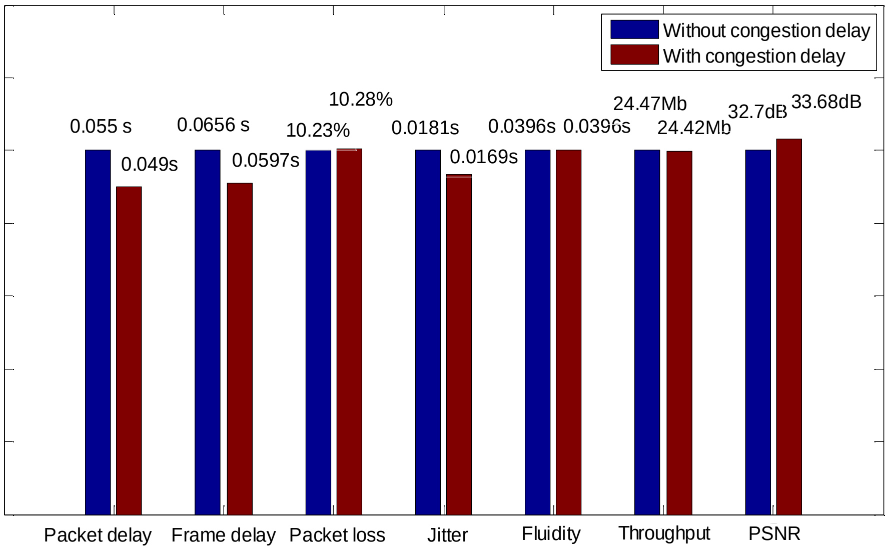
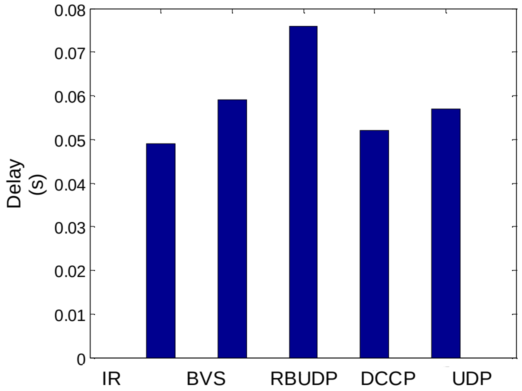


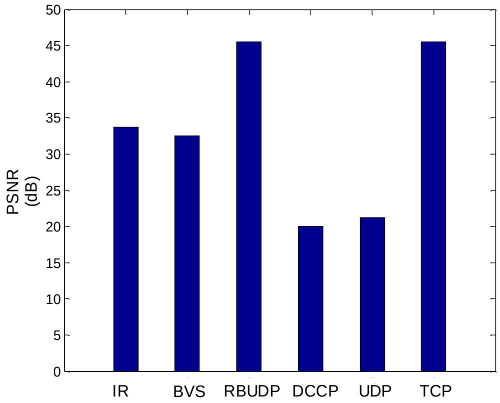
Figure 5.9: Congestion delay impact

5.2.8 Performance comparisons

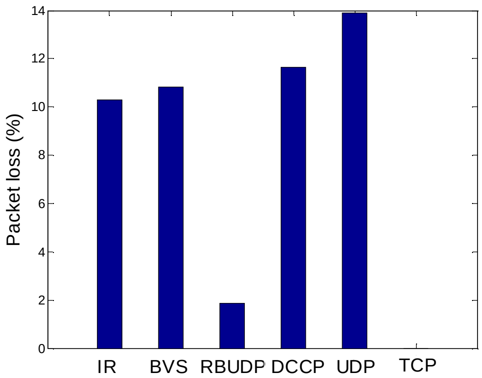
Figure 5.10 shows the performances of the examined protocols. QR retransmission protocol is represented by BVS protocol. Unlike UDP, BVS and RBUDP, the proposed protocol reacts to network congestion by postponing the next packet transmission. This response helps the proposed protocol to reduce queue and suppress end-to-end delay.



(a) Delay



(b) PSNR



(c) Packet loss

Figure 5.10**:** Performance comparisons of transport protocols

Conversely, although DCCP and TCP implement congestion control, these protocols require certain observation periods before reducing or increasing the transmission rate. DCCP requires a feedback packet containing receiver observations, while TCP implements a time-out before detecting network congestion.

By arranging retransmission time and quickly responding to packet loss, the proposed protocol successfully reduces packet and frame delays. TCP and RBUDP experiences significant packet and frame delays (TCP delay is not shown in Figure 5.10a).

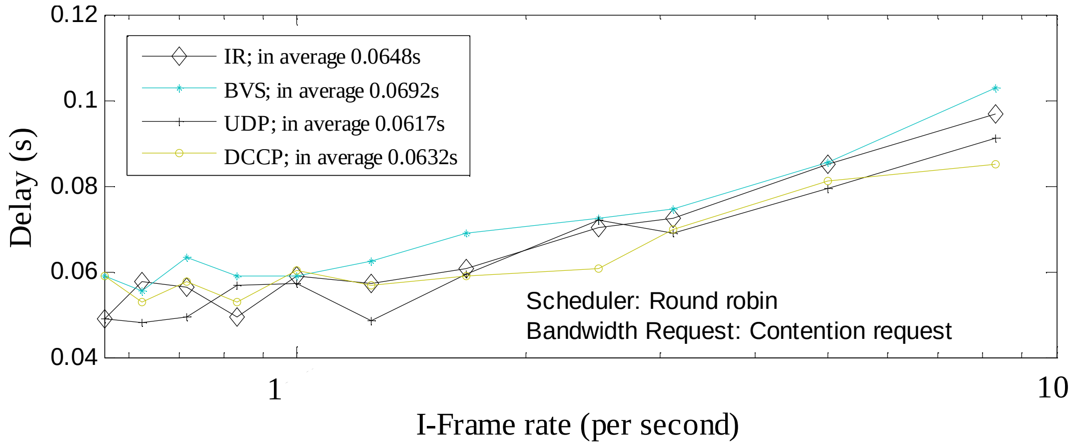
In comparison with UDP, DCCP and BVS, IR reduces packet loss significantly. The loss is 3.5% lower than UDP, 1.37% lower than DCCP and 0.56% lower than BVS. Therefore, the proposed protocol has higher throughput than those protocols.

Furthermore, IR is able to preserve priority packets better than BVS, which also retransmits priority packets. Consequently, IR produces better video quality as shown in Figure 5.10c. Although IR has higher packet loss and lower PSNR value than TCP and RBUDP, its delay characteristics are more desirable for real-time video transmission.

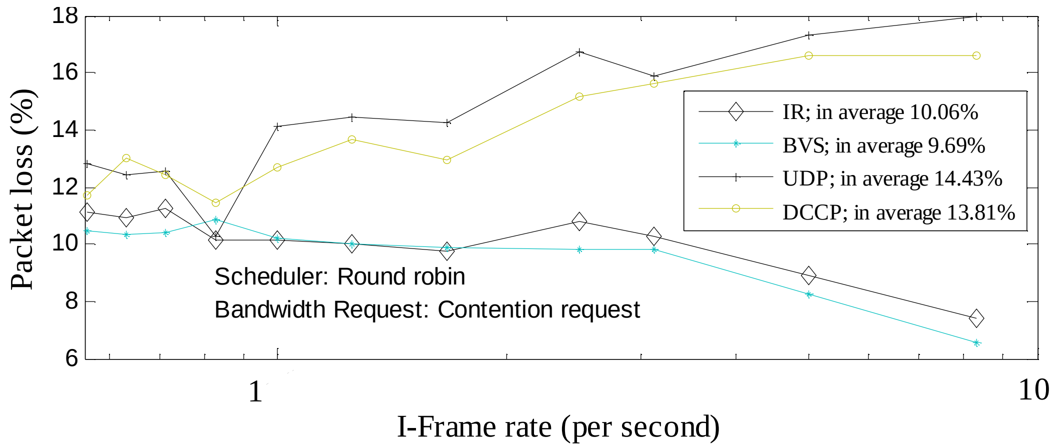
The I-frame rate term is used to show how often the prioritized packets (I-Frames) occur within video data. These prioritized packets require retransmission if packet loss is detected. The increasing frame rate also represents video bit rate increment as I-frames contain more bits.

IR has better packet delay and packet loss performances than the existing the unreliable retransmission protocol BVS. Although DCCP delay is lower than the proposed protocol, its video quality performance, PSNR, is much lower than IR.

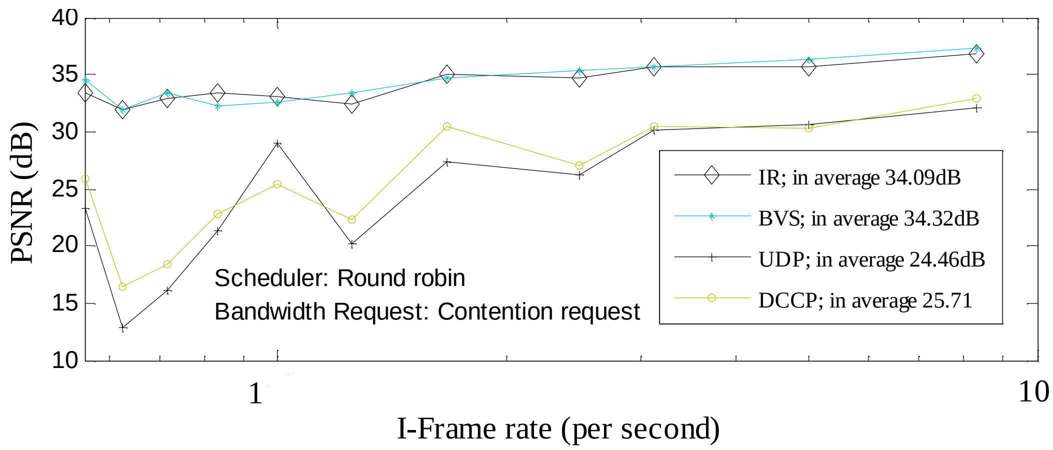
Figure 5.11highlights that all protocols experience increasing delay when I-frame rate increases. Since TCP and RBUDP experience high delay, neither is shown in Figure 5.11. In general, UDP delay is superior to other protocols; however, it experiences more losses than the others; consequently, the average PSNR of UDP is the lowest.



(a) Delay



(b) Packet loss



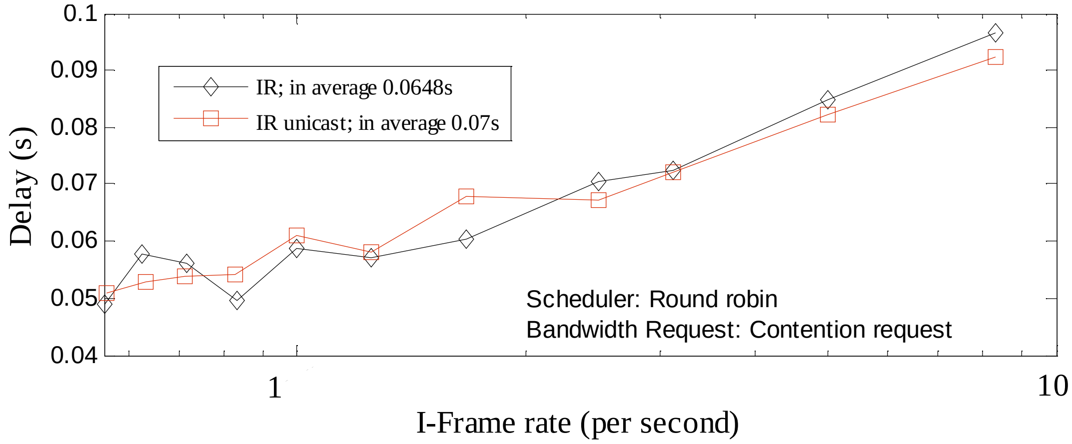
(c) PSNR

Figure 5.11: Performance comparisons

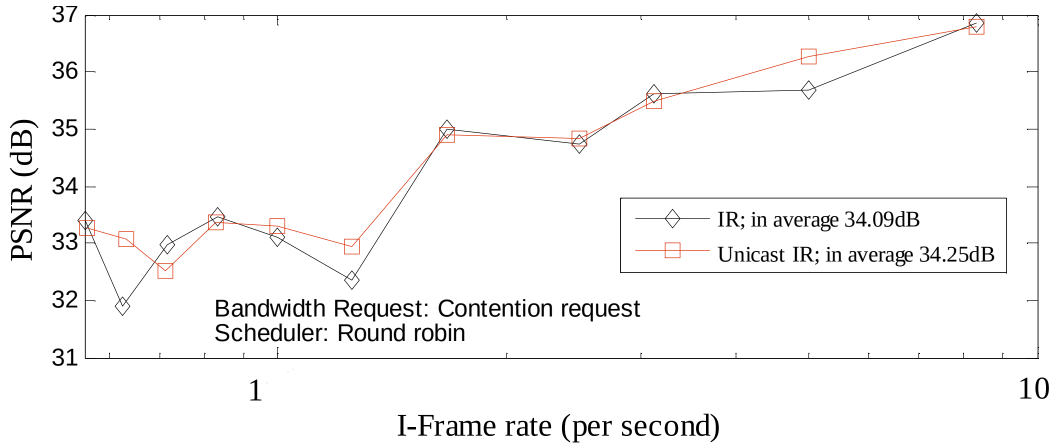
5.2.9 Unicast CD performance

Figure 5.12 shows that unicast CD successfully reduces IR packet delay and increases PSNR for I-frame rate higher than 2 frames per second. However, for frame rates lower than 2 frames per second, the average delay and PSNR are lower than the original IR. This shows that unicast CD works effectively when the I-frame rate is high.

As unicast CD does not perform well for low retransmission rate, the method is optional for frequent retransmission.



(a) Delay



(b) PSNR

Figure 5.12: IR with unicast CD performance

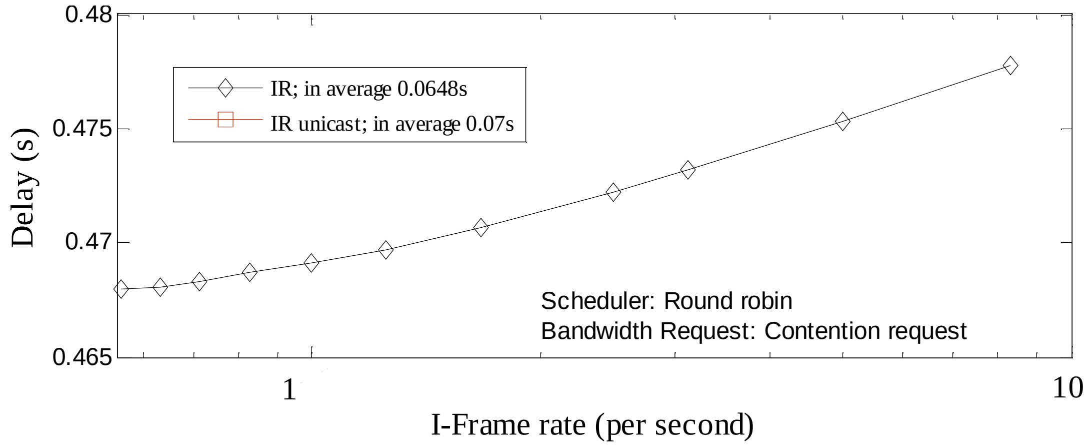


Figure 5.13: TCP performance

TCP performs poorly and produces significant delay when delivering surveillance videos over WiMAX. Figure 5.13 shows that TCP yields extremely high packet and frame delay when mobile nodes use higher video rates.

5.2.10 Inter-frame retransmission in other network (802.11)

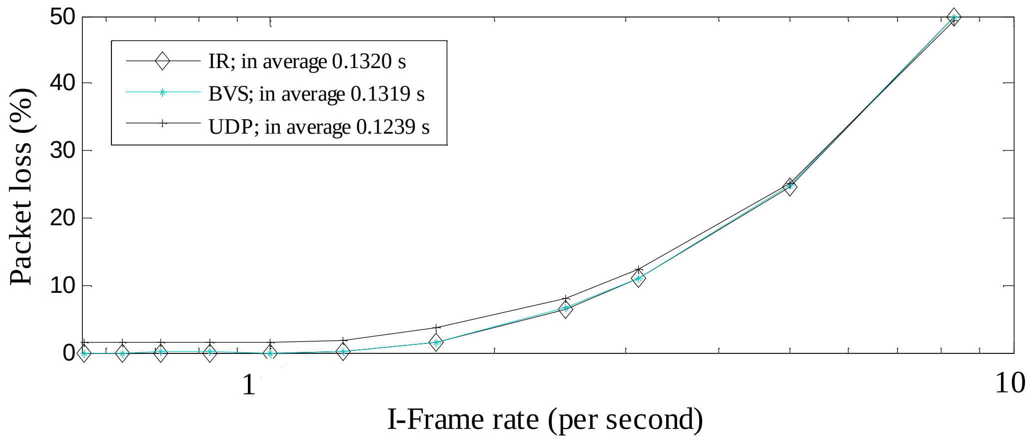
Although this thesis discusses WiMAX-based surveillance network, this thesis also evaluated the proposed inter-frame retransmission protocol in 802.11 environments. This is important as the proposed protocol, so far, is independent of hardware: PHY and MAC layers. Inter-frame protocol can also be implemented in a 802.16 environment. The simulation scenario, video trace and evaluation tools are similar to those outlined in Section 2.7, except the network is 802.11.

The performance of the proposed protocol in terms of packet loss percentage can be observed in Table 5.1. Although irregularity occurs when GOP 3 (I-frame rate 8.33/s), UDP suffers high packet loss in all other GOPs. The comparison of the loss percentage between BVS and IR reveals the superiority of the IR protocol. Table 5.1 is redrawn in Figure 5.14a.

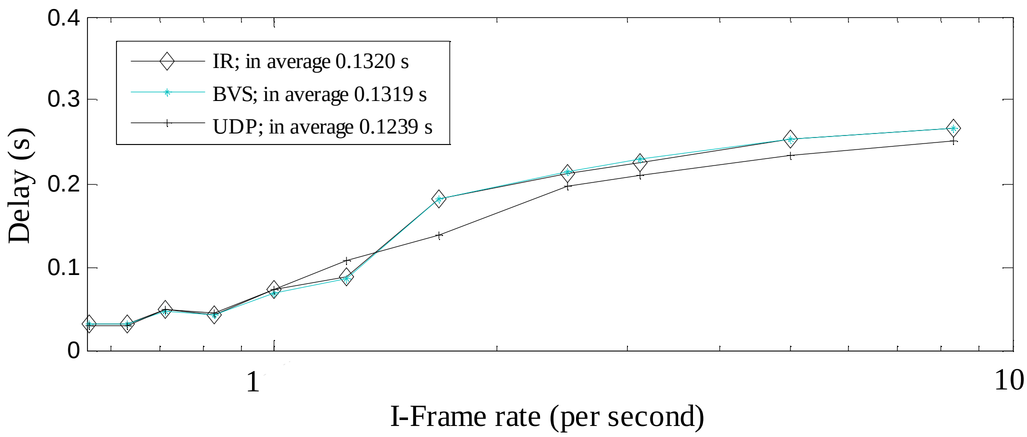
Table 5.1: Packet loss comparison

Protocol	3	5	8	10	15	20	25	30	35	40	45
UDP	49.45	25.27	12.25	8.00	3.61	1.73	1.51	1.48	1.48	1.48	1.48
BVS	49.93	24.79	11.14	6.55	1.59	0.16	0.00	0.07	0.13	0.00	0.00
IR	49.86	24.71	11.04	6.40	1.42	0.16	0.00	0.00	0.00	0.00	0.00

The average delay versus I-frame rates for UDP, BVS and IR are depicted in Figure 5.14b. For I-frame rate > 1.25, UDP experiences the lowest delay; however, delay patterns are irregular when I-frame rate ≤ 1.25. This irregularity is caused by the video rates are relatively lower than the available bandwidth, so that the protocols experience almost similar delay. In contrast, delay differences between BVS and IR are not significant. Both protocols experience an almost similar delay around 0.132s.



(a) Packet loss



(a) Packet loss

Figure 5.14: Packet loss and delay

5.3 Transport and MAC cross-layer protocol

The proposed protocol in the previous section is able to increase the existing NACK-based retransmission protocols. However, as shown in Section 5.2.8, its performance is not consistent with all I-frame rates. This is caused by the retransmitted packet(s) increased the bandwidth consumption. Although CD suppresses the retransmission effect, the improvement is not major; a risk to traffic fluctuations is still present.

This section addresses the traffic increment problem caused by retransmission in IR protocols. The additional bandwidth is required to allocate the retransmitted packet(s). However, the bandwidth increment cannot be performed in network layer. The only way is by utilizing MAC layer support.

The MAC support for transport layer protocols has been used explicitly in some existing cross-layer protocol, as mentioned in Section 3.5. Although those MAC assisted transport layer protocols in [115, 116] perform better than the basic ones, the reliability of transport layer protocols with congestion control is not suitable for real-time video transmission as those protocols continue to exert tremendous delay.

This section introduces the MAC assisted transport layer through an early bandwidth request mechanism to accommodate the retransmitted packet. The MAC assistance is applied to the previous IR protocol. It is important to note that the proposed protocol is intended only for UL video streaming in the WiMAX network.

The proposed MAC assisted transport layer protocol or Transport – MAC cross-layer protocol combines IR and bandwidth allocation in WiMAX to achieve better video surveillance performance. The transport and MAC cross-layer (TMC) term refers to a method that explores interactions between the MAC layer and the transport layer. The proposed method aims to enable the MAC layer to support retransmission in the transport layer.

The two-way interactions are avoided to prevent processing delay. Instead, the proposed method enables the MAC layer to read the transport layer header in order to provide a service to the transport layer. The service is an early bandwidth request to accommodate the retransmitted packet(s).

Besides improving the IR performance, the proposed method requires only a minor change in the MAC layer so that it is applicable to existing WiMAX devices and compatible with other MAC functionalities.

5.3.1 Early bandwidth request

When an SS receives a NACK packet requesting retransmission, MAC layer should be able to determine how many packet(s) are requested. Instead of waiting for the retransmitted data available in MAC buffer and the scheduler requests bandwidth based on MAC buffer capacity, the proposed early bandwidth request directly to add the retransmitted bytes. If there is a bandwidth request for other data preceding the retransmitted one, the retransmitted traffic is embedded in this request. No separate bandwidth request is required.

Figure 5.15 illustrates the early bandwidth request implementation in NIST WiMAX module [40]. The frame re-assembler in the MAC layer reads the NACK packet and notifies the scheduler to add the amount of the requested bytes ($Bytes_{NACK}$) in the bandwidth request. In turn, the scheduler sends the bandwidth request based on data size on MAC buffer ($Bytes_{BUFFER}$) and the retransmission request ($Bytes_{NACK}$).

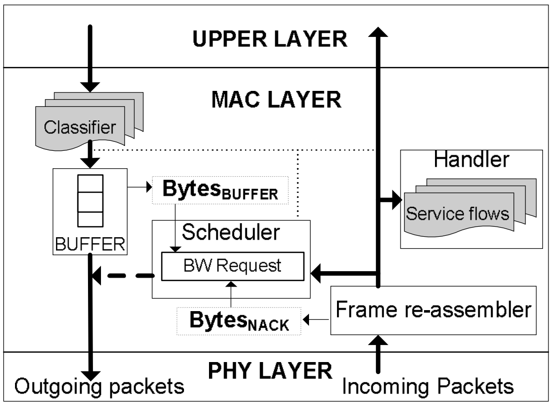
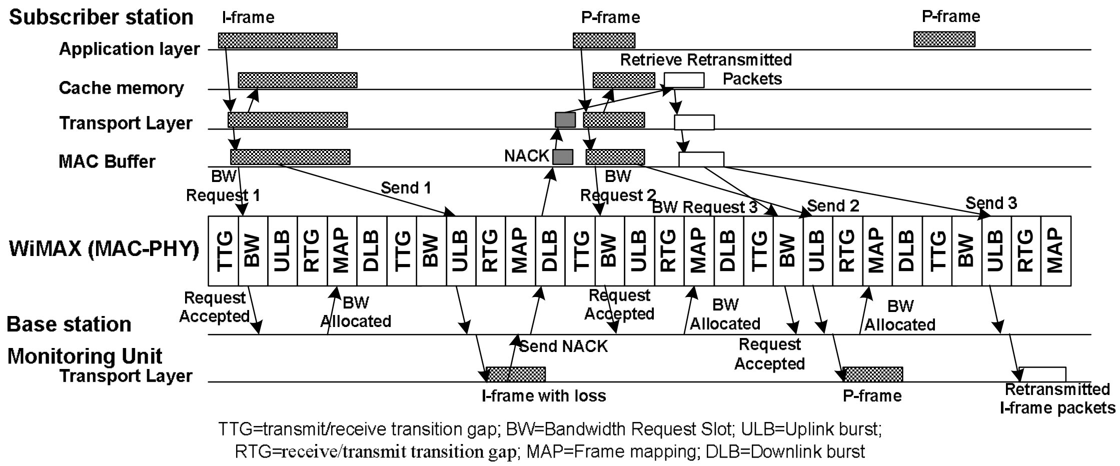


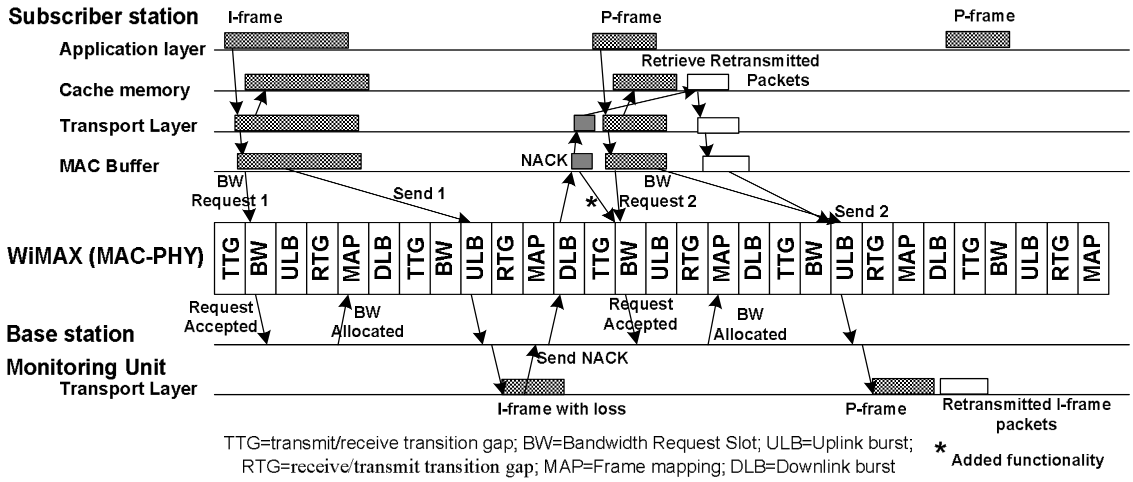
Figure 5.15: MAC layer implementation

Figure 5.16 shows the comparison of simplified layer interactions of the IR protocol with and without MAC support. In Figure 5.16a, the bandwidth for the retransmitted packets is requested separately (request 3) as the packets are not available by the time the SS sends a bandwidth request (request 2) to BS. Consequently, instead of sending the retransmitted data in the nearest UL burst (send 2), SS will allocate it after the next burst (send 3). This postponement increases the packet delay.

On the other hand, the MAC cross-layer accelerates retransmission as the earliest bandwidth request accommodates the retransmitted packets. MAC layer reads NACK packet and adds the byte amount to the nearest bandwidth request ('*' in Figure 5.16a). The retransmitted and the current data are sent within the same burst (send 2).



(a) Inter-frame retransmission without MAC cross-layer



(b) Inter-frame retransmission with MAC cross-layer

Figure 5.16: Subscriber station based MAC cross-layer

5.3.2 The impact of MAC cross-layer

Transport layer packets queue in the MAC buffer of the SS before being transported by the physical layer. MAC transfers the data to the allocated time in an UL-sub-frame based on the duration allocated by BS in UL-MAP. The duration itself is decided by BS based on SS bandwidth request and the available bandwidth. Since the main feature of the MAC cross-layer is additional bandwidth for the retransmitted packets, the proposed protocol obtains higher bandwidth allocation than the basic IR protocol.

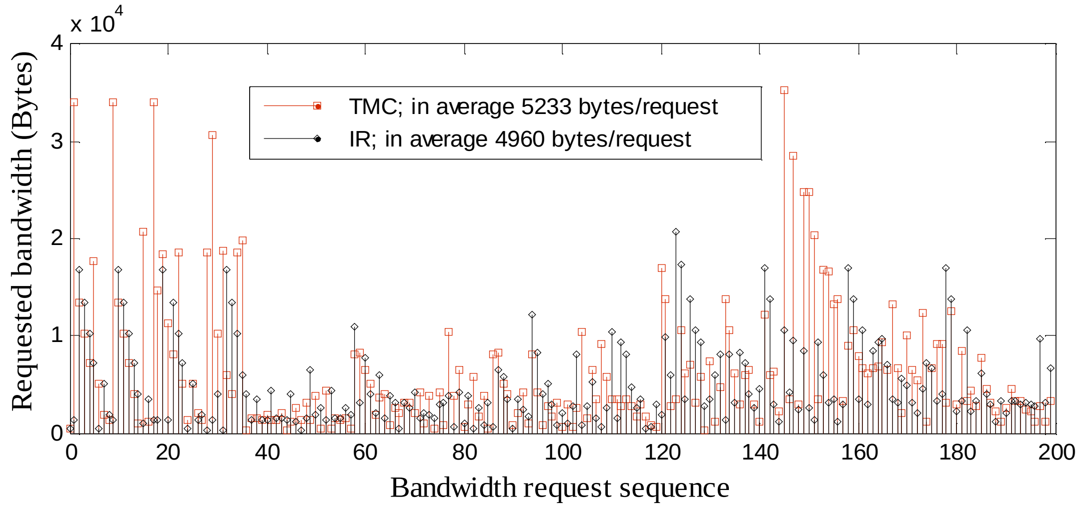
Table 5.2: Bandwidth comparison for GOP 30

Protocol	IR	TMC
Number of bandwidth requests	1270	1268
Average requested bandwidth	4960	5233
Number of uplink transmission	1530	1522
Average allocated bandwidth	2419	2430
Network utility	55.29%	55.54%

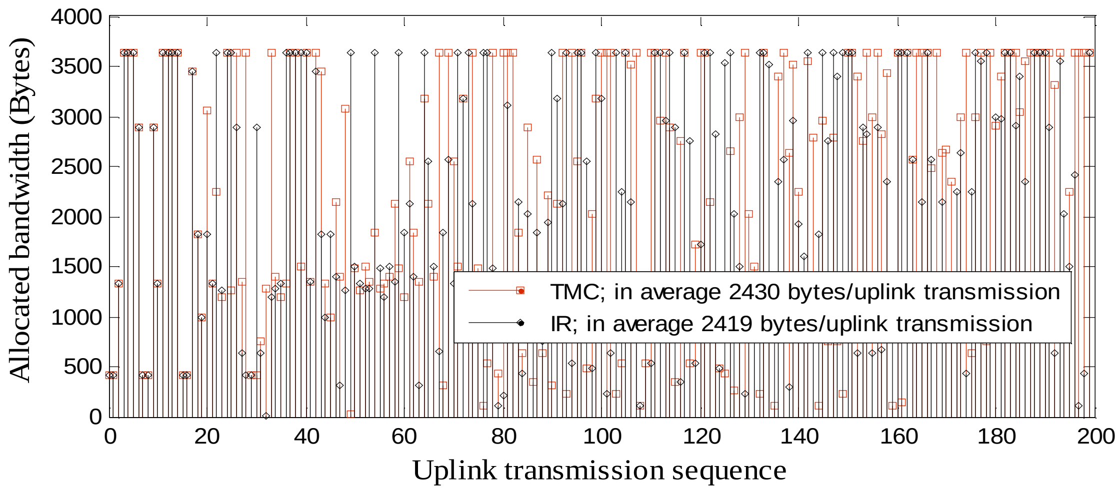
For the simulated traffics outlined in Section 2.7.7, TMC generates 1268 times bandwidth requests, while IR produces 1270 times. These values show that TMC minimizes the number of bandwidth requests. As shown in Table 5.2, TMC requested 5233 bytes/request in average, 273 bytes higher than IR. From those requests, BS allocates in average 2430 bytes/uplink transmission for TMC and 2419 bytes/uplink transmission for IR.

TMC utilizes network better than IR. Since frame duration is 5ms and maximum network throughput (Figure 2.11a) is 7Mbps, TMC has network utility=(2430x8/0.005)/7000000x100%=55.54%. IR utility is only 55.29%, lower than TMC. Figure 5.17 shows the requested and allocated bandwidth for both protocols for the first 200 bandwidth requests.

Since the additional bandwidth is requested before the retransmitted packets available in MAC buffer, the allocated bandwidth can be used by regular data, even if the retransmitted packets fail to be retrieved. The higher bandwidth allocation and network utility in the proposed protocol produce lower delay and higher video quality.



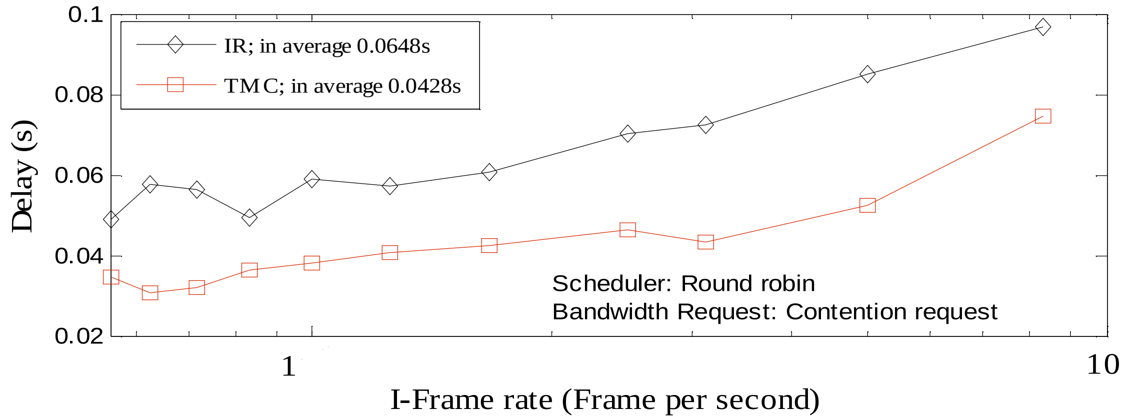
(a) Requested bandwidth comparison



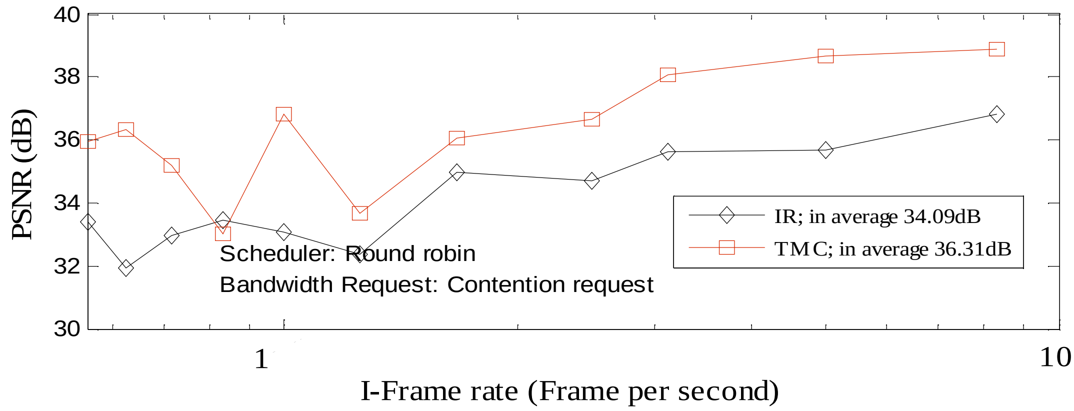
(b) Allocated bandwidth comparison

Figure 5.17: The requested and the allocated bandwidth for GOP 30

Figure 5.18 shows the performance comparison between the original IR protocol and the proposed transport – MAC cross-layer protocol. TMC consistently reduces packet delay for all I-frame rates. Although PSNR is dropped when sending data with I-frame rate 0.833 fps, it is probably caused by the subsequent undecodable error frames.



(a) Delay



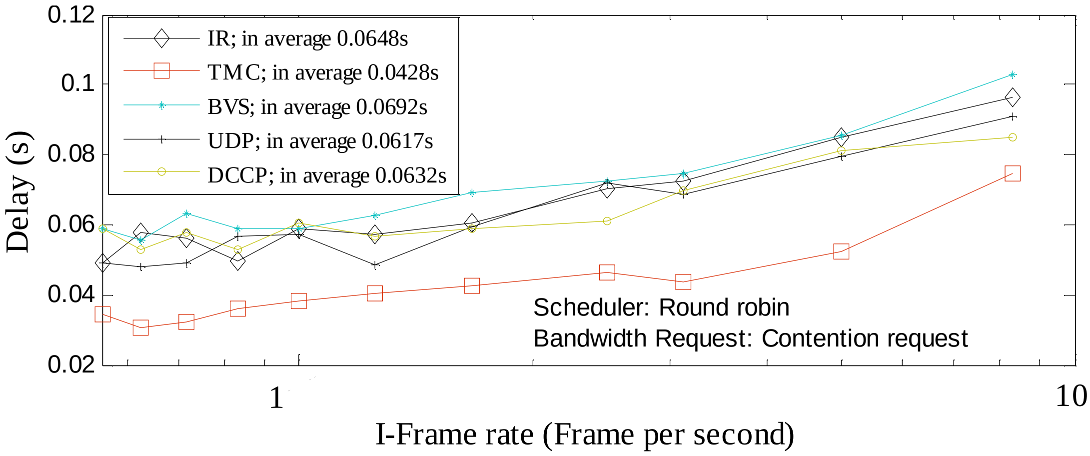
(b) PSNR

Figure 5.18: Performance comparison between IR and TMC

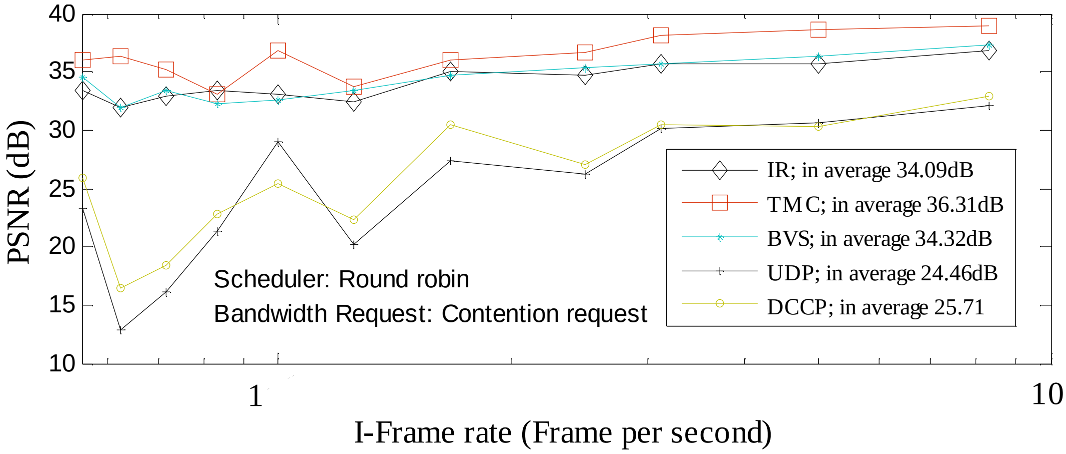
5.3.3 Transport protocol comparisons

TMC was compared to the existing protocols, as shown in Figure 5.19. TMC outperforms the existing protocols, including UDP. TMC is able to reduce UDP delay up to 18-37%. TMC achieves PSNR improvements around 14.3-149.5%, 12.6-150.2%, 21.3-184.3% and 17.9-120.23% over IR, BVS, UDP and DCCP respectively.

Other existing protocols, such as SCTP and RBUDP, are not presented as they are compared in [130]. The result proves that TMC outperforms the existing protocols for surveillance application over WiMAX with uniform traffic.



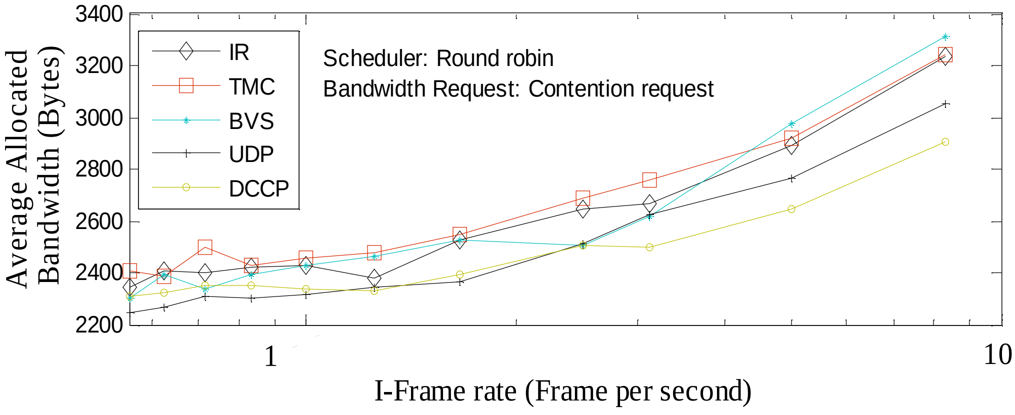
(a) Delay



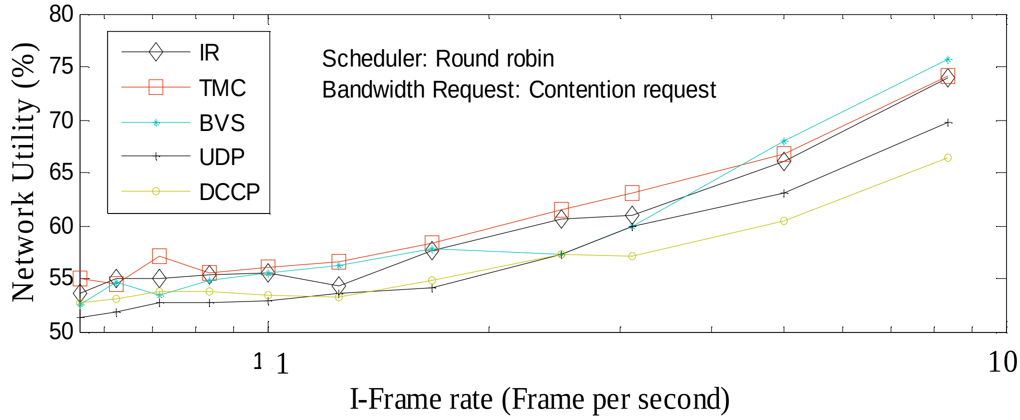
(b) PSNR

Figure 5.19: Performance comparison between TMC and other protocols

TMC produces lower delay than UDP because TMC requests more bandwidth when packet loss occurs. As shown in Figure 5.20, TMC received more bandwidth allocation than other protocols and experienced lower allocation than BVS for high I-frame rates as maximum network throughput (Figure 2.11a) limits bandwidth allocation for the retransmitted packets. However, the limited bandwidth does not reduce TMC performance as the cross-layer still yields better bandwidth allocation.



(a) Average allocated bandwidth



(b) Network utility

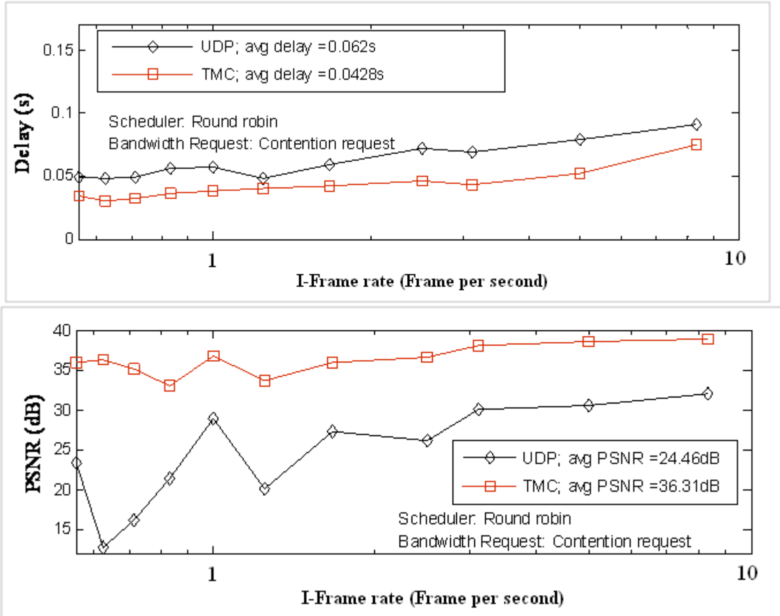
Figure 5.20: Comparison of the allocated bandwidth and network utility

On the other hand, although BVS received higher bandwidth allocation for high I-frame rates, bandwidth may be wasted as multiple NACK in BVS may disturb regular packet transmission. UDP and DCCP suffer lower bandwidth allocation as both protocols do not retransmit packet loss. UDP does nothing to increase network utility but it may experience separated frame transmission more frequently than TMC, which causes higher delay.

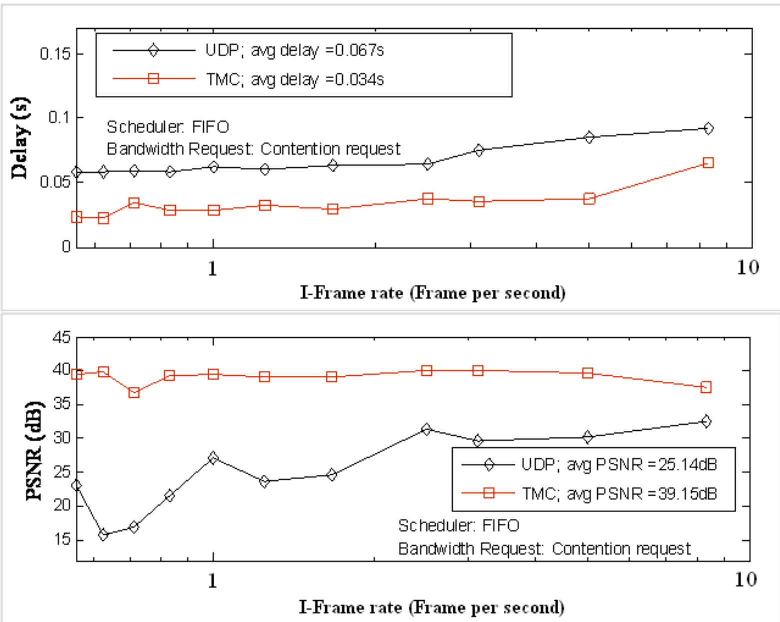
5.3.4 Protocol performance over various schedulers

In order to ensure that the proposed protocol is suitable for various WiMAX schedulers, the protocol was evaluated on RR, FIFO, frame-based and the well-known EDF schedulers for dedicated video surveillance over WiMAX. The results show that TMC is able to reduce the delay and improve the PSNR in almost all schedulers (Figure 5.21).

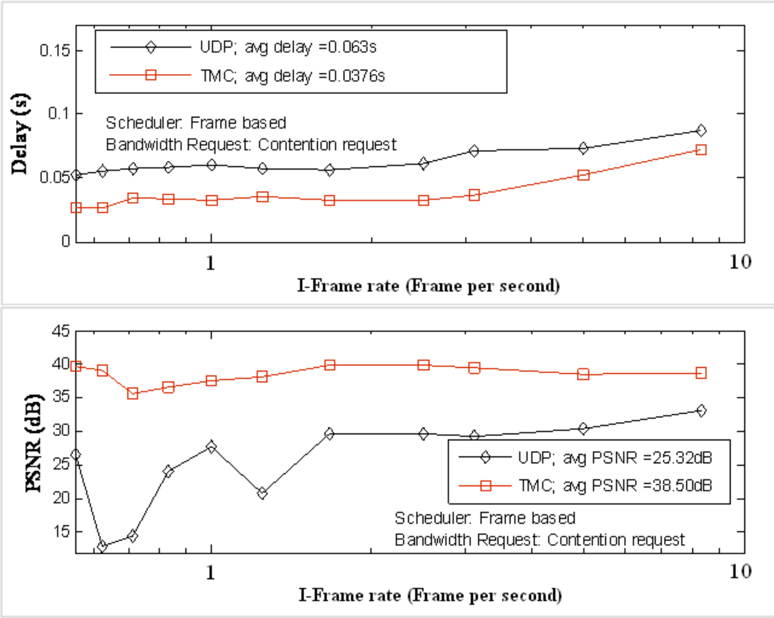
The RR, FIFO and frame-based schedulers experience significant delay reductions and PSNR improvements.



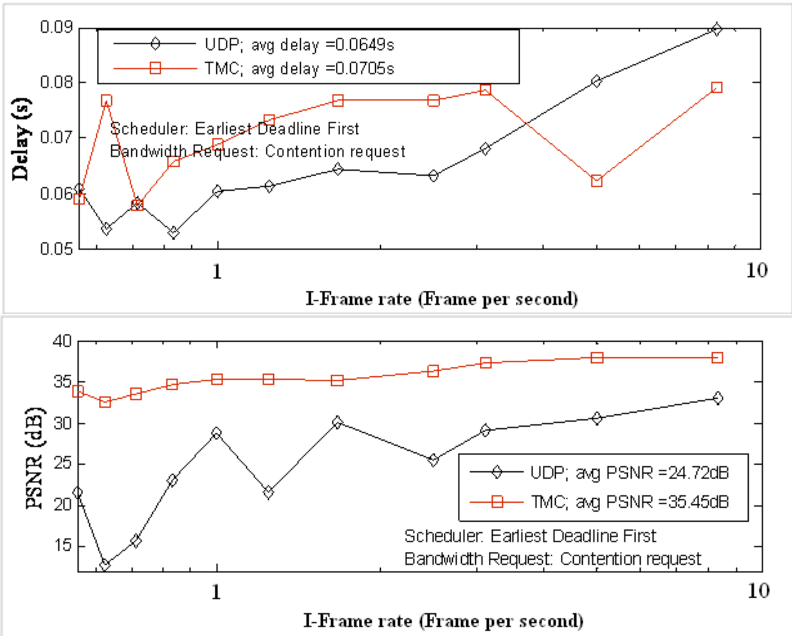
a. Round robin scheduler



b. FIFO scheduler



c. Frame based scheduler



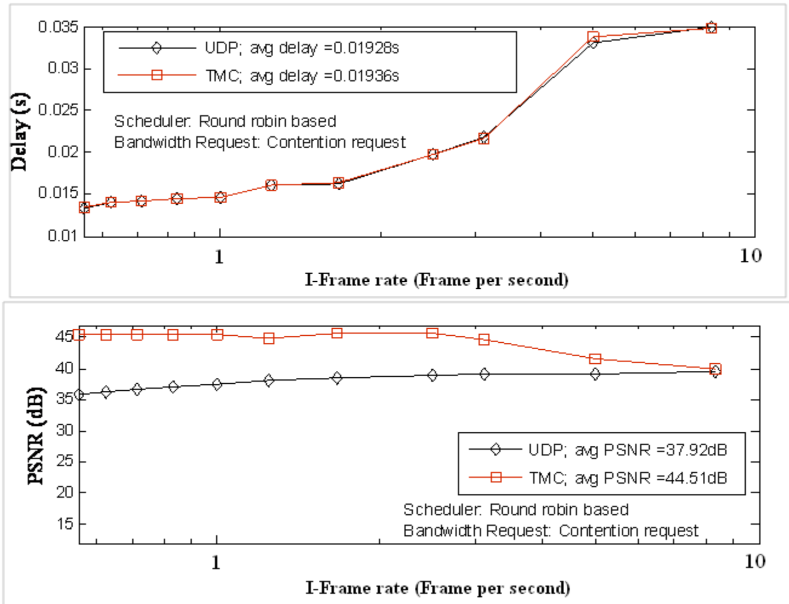
d. EDF scheduler

Figure 5.21: TMC performances over various schedulers

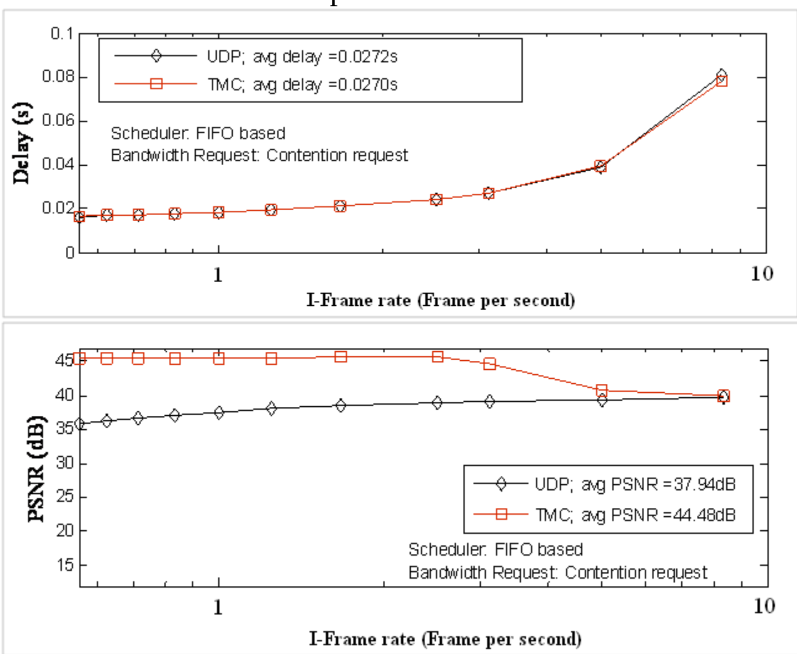
TMC does not work well in the EDF scheduler as its delay varies between EDF and UDP. The reason is that the EDF scheduler is not suitable for dedicated video surveillance with uniform traffics over WiMAX, as the traffic has similar behaviour and deadlines, while the EDF scheduler classifies the allocated data based on traffic deadline. As a result, BS performs unnecessary sorting and imposes unnecessary delay. Although TMC failed to reduce delay in all rates, it increases PSNR consistently.

5.3.5 Protocol performance over previously proposed schedulers

TMC is able to improve PSNR performance on the previously proposed schedulers without overtly affecting the delay (Figure 5.22). On average, TMC improves PSNR values 6.59dB and 6.54dB for RR and FIFO-based schedulers respectively.



a. RR-based packet-aware scheduler

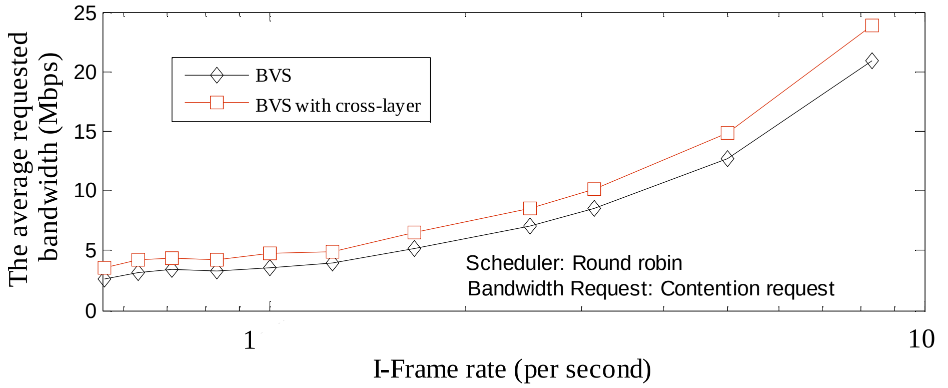


b. FIFO-based packet-aware scheduler

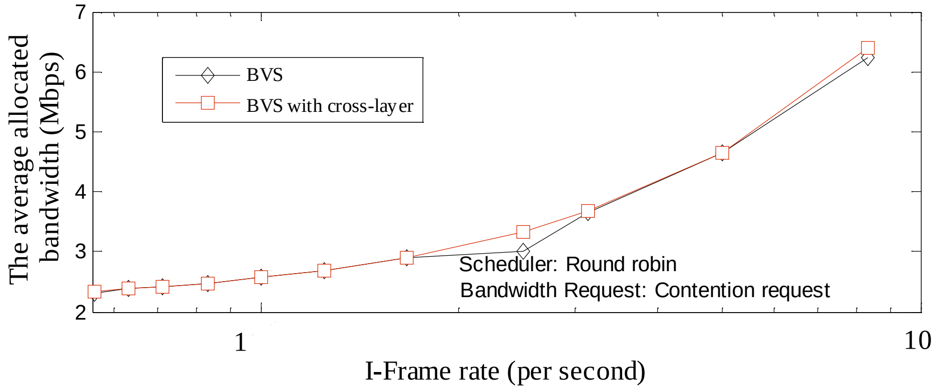
Figure 5.22: TMC performances for packet-aware schedulers

5.3.6 Cross-layer impact to other protocol

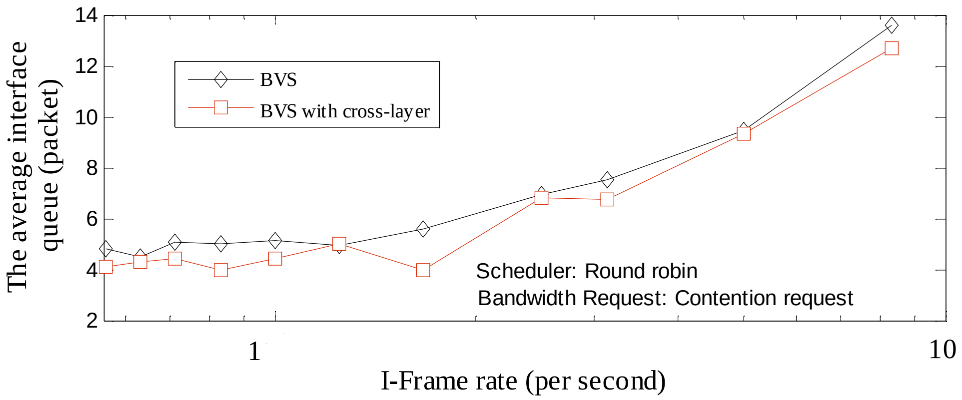
The cross-layer was implemented in another retransmission protocol; BVS [102]. By using the simulated scenario outlined in Section 2.7.7, the results were presented in Figure 5.23.



(a) The requested bandwidth



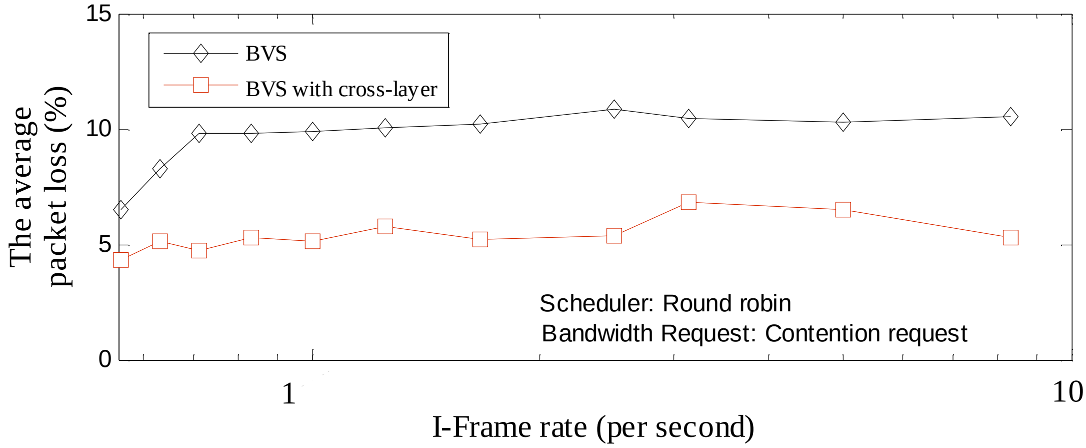
(b) The allocated bandwidth



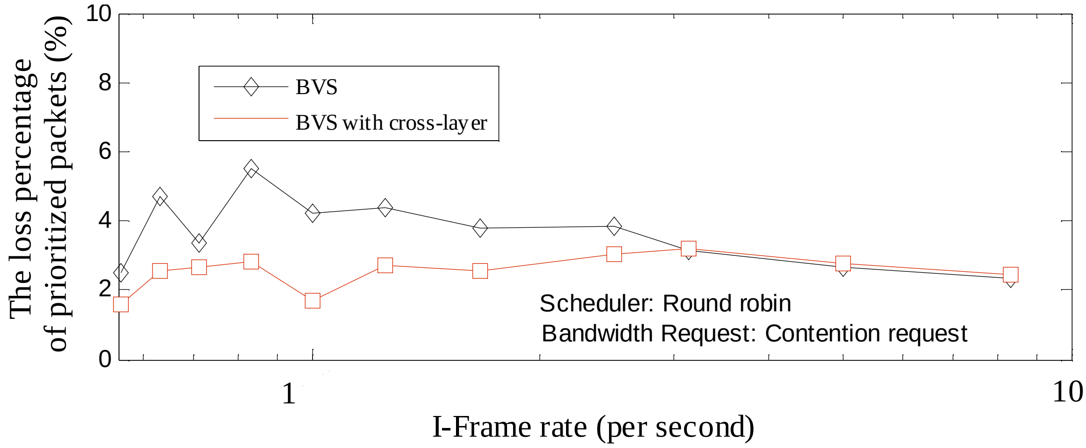
(c) Interface queue

Figure 5.23: Cross-layer impact on bandwidth and buffer

The method increases the requested bandwidth to BS, as shown in Figure 5.23a. The requested bandwidth of BVS increases in all video rates when the cross-layer method is applied. The average allocated bandwidth pattern (Figure 5.23b) does not follow the request pattern as the bandwidth allocation is limited by the maximum network throughput (Figure 2.11). The allocated bandwidth increases only slightly, however, the method reduces successfully buffer load (interface queue (IFQ), Figure 5.23c.). IFQ reduction avoids packet loss caused by buffer overflow. Moreover, IFQ load reduction shows that more packets are transported successfully than without the cross-layer schema.



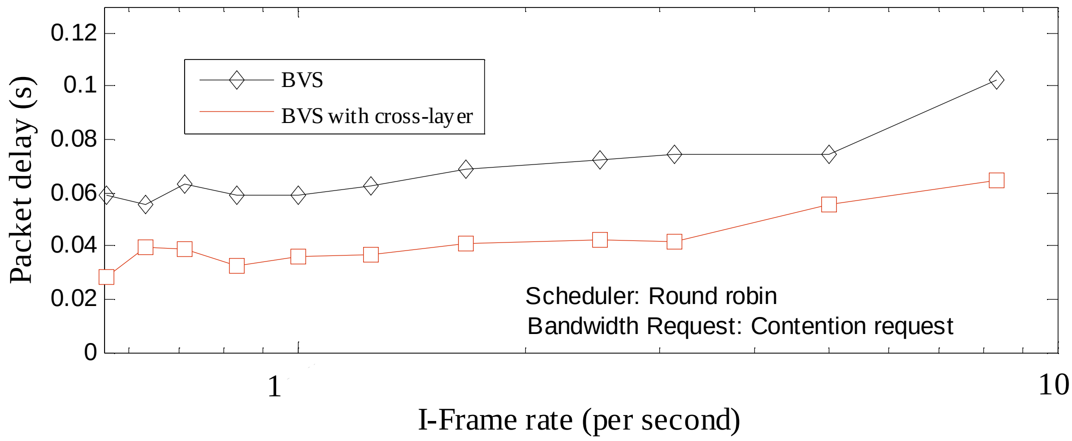
(a) Packet loss



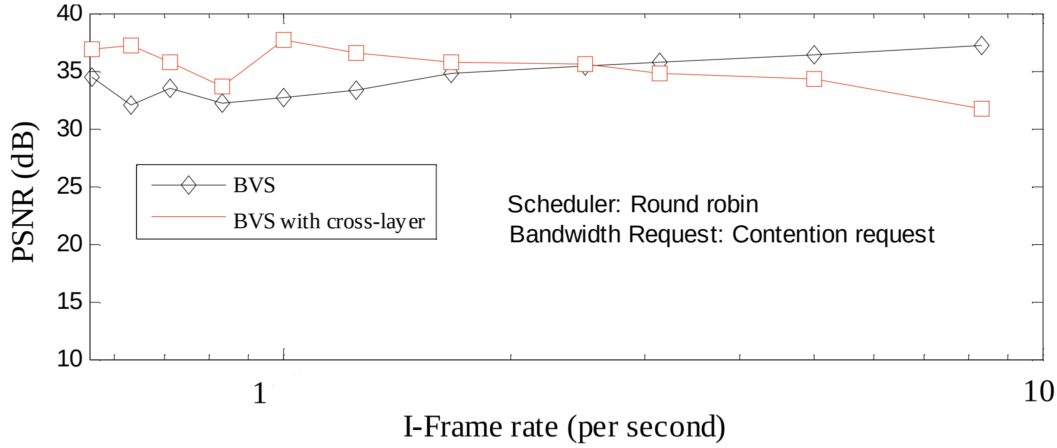
(b) Loss of prioritized packet

Figure 5.24: Cross-layer impact on packet loss

IFQ reduction in SS avoids buffer overflow and reduces the volume of packet loss. As depicted in Figure 5.24a, the proposed cross-layer reduces significantly the average packet loss up to 4.27%. The average packet loss decreases in all I-frame rates. Since video packets contain I-frames and P-frames, the packet-loss distribution among I-frames and P-frames is not similar. The higher I-frame rates the higher retransmission frequency. The frequent retransmission and bandwidth limitation cause I-frame experiences more loss than P-frames. As shown in Figure 5.24b, losses of prioritized packets are slightly higher when I-frame rates are greater than 3.



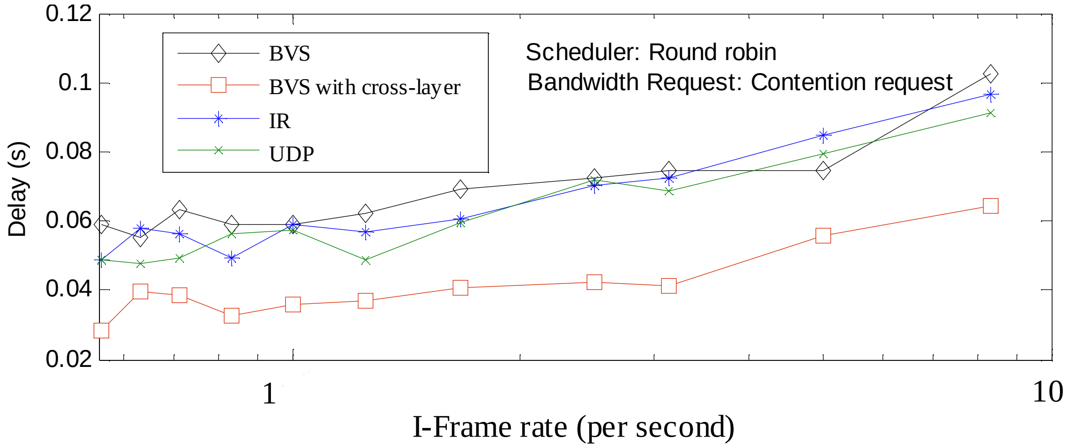
(a) Delay



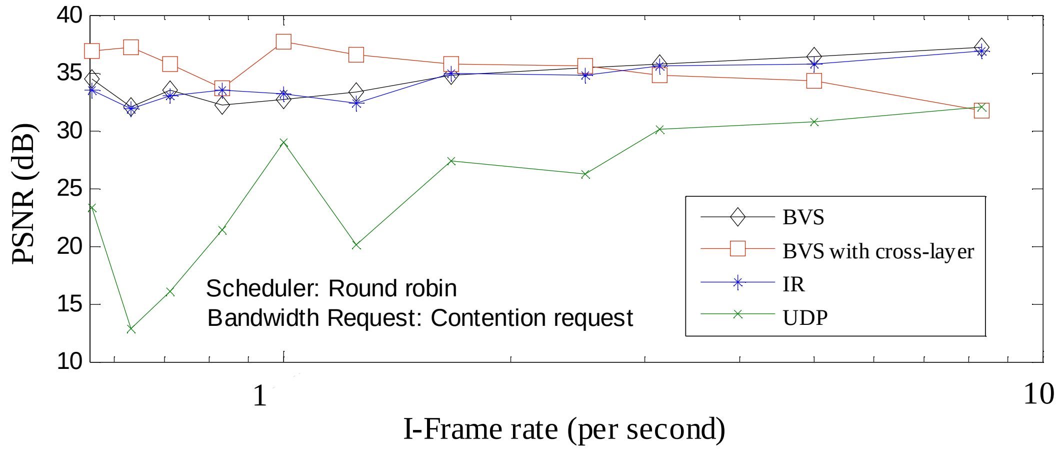
(b) PSNR

Figure 5.25: Cross-layer impact on delay and PSNR

Figure 5.25 shows the impact of cross-layer improvement on delay and PSNR. MAC cross-layer reduces delay and improves PSNR significantly. Delay reduction occurs as cross-layer provides early bandwidth request and allocates more bandwidth than the original protocol. While delay is reduced consistently for all I-frame rates, PSNR values decrease when I-frame rates are greater than 3.15. These were caused by very frequent packet retransmission, which generates excessive additional bandwidth (Figure 5.23a) and leads to high dropped packets in the prioritized packets (Figure 5.24b). The cross-layer decreases the BVS delay 26.7 ms in average and increases PSNR up to 5.13 dB.



(a) Packet delay



(b) PSNR

Figure 5.26: Protocol performance comparisons

The proposed method was also compared to some existing reliable protocols, as shown in Figure 5.26. The method outperforms the existing protocols. The proposed cross-layer has a lower delay than UDP because the early bandwidth request not only affects the retransmitted packets, but also reduces the queue in the SS buffer. Since UDP does not response to congestion, it can make it even worse. UDP experiences the lowest PSNR, as it ignores the dropped packets. The proposed method reduces the BVS delay by 39.3% on average; 35% better than IR and 32.3 % lower than UDP. It improves BVS PSNR up to 5.13 dB, up to 5.21 over IR and on average, 10.9dB over UDP.

5.4 Summary

This chapter proposes IR and TMC protocols to reduce packet loss in video surveillance over a WiMAX network.

The prioritized IR scheduling with CD method within IR protocol is able to make the proposed protocol perform better than existing protocols such as BVS, DCCP and UDP.

The TMC protocol utilizes MAC layer support to provide additional bandwidth for the retransmitted packet. The proposed TMC protocol achieves a more consistent improvement than IR protocol. The early bandwidth request method is proposed to avoid delay caused by bandwidth request mechanisms.

The early bandwidth request method requires MAC to be able to read the content of NACK packet and add the amount of requested bytes to the earliest bandwidth request. The proposed cross-layer technique is able to improve significantly the IR protocol, decreasing packet delay and increasing the quality of the received video. The proposed TMC also works very well on various schedulers, including those proposed in Chapter 4.

The cross-layer scheme was also applied to the existing BVS protocol. The simulation results show that the method is able to enhance significantly the BVS protocol performance.

Chapter 6
Application layer: redundancy and error concealment

6.1 Introduction

Video surveillance is usually implemented in a well-managed environment, where proper adjustment is made during initial installation. However, wireless environment irregularity, multipath transmission, shadowing and interferences produce signal loss as well as packet errors. These conditions lead to performance degradation.

An application layer technique is the last opportunity to overcome flaws in wireless transmission. As mentioned previously in Section 2.6, application layer techniques are classified into three levels: encoding, streaming and operating system. This thesis focuses on the streaming techniques that consist of active and passive methods. Active streaming is reactive to network change, but the passive method is not.

The WiMAX-based video surveillance network is a managed network where bandwidth requirement for each connection is well planned and connectivity is guaranteed. Network adaptation to traffic characteristics has been accommodated in MAC and transport layer as discussed in Chapters 3 and 4. Therefore, this chapter focuses on passive streaming techniques: redundancy and error concealment techniques. Application layer redundancy is performed by implementing the rateless code [131], while the error concealment techniques are implemented following the decoding process.

6.2 Redundancy techniques

Wireless video transmission is sensitive to interference. The channel introduces signal degradation and packet errors. In order to reduce the effect of the error-prone environment to video quality, redundancy was implemented optionally. The implementation is optional because, if the transmission link is good, excessive solutions to tackle high packet loss may reduce transmission performance.

In communication systems, redundancy can be implemented in all layers with different forms. For example, the space antenna diversity is physical layer redundancy and cyclic redundancy is MAC layer redundancy. Therefore, redundancy was implemented selectively. The redundant packets are added only for prioritized packets, since they play an important role in video performance. Redundancy can take the form of either transport-layer redundancy or application-layer redundancy.

6.2.1 Transport-layer redundancy

Video frames are assumed to have a sequence: { F_0 F_1 F_2…}, where F_i=I or P frames and i=0,…, N-1, N is the number of the transmitted frames. Suppose that p_k is packet sequence of frame F_i where k=0,…, M-1. M is the number of F_i packets, that is obtained from the ceiling of F_i size divided by packet size. Then the transmitted frames: {F_0 F_{0r} F_1F_{1r} F_2 F_{2r}…}, where redundant frame $F_{ir}\neq0$ if F_i=I frame. The redundant packet sequence is $p_{kr}=p_k$. Figure 6.1 illustrates the packet redundancies.

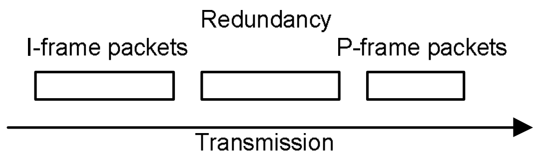


Figure 6.1: Packet redundancies

6.2.2 Application layer redundancy

The weakness of transport-layer redundancy is that the receiver may receive duplicate packets, which is a waste. The redundancy has one-to-one correlation and is not able to contribute to other packet recovery. Furthermore, the transport-layer protocol may experience processing overheads as it should store the prioritized transmitted packets and resend them later. Therefore, redundancy was implemented with many-to-many correlation, which is implemented in addition to transport layers.

Rateless codes have attracted researchers for their performances working on the erasure channel. Among the codes, the Luby transform [131] was the well-known first practical realisation of rateless codes. If it is the original data, then redundancy (j) is added and k=i+j data is transmitted. The code is then decoded based on the amount of data received (m). The failure probability (Pf) of the rateless code decoding is provided by [131]:

$$P_f(m,k) = \begin{cases} 1, & if\ m < k \\ 0.85x0.567^{m-k}, & if\ m \geq k \end{cases} \quad (6.1)$$

Work on [132] suggested piggybacking extra redundancy for previous packets. However, such methods introduce additional processing overhead, as well as packet delay. Since the network has a bandwidth constraint, this thesis implements full redundancy for selected/prioritized data.

6.2.3 Redundant packets transmission

Transmission time for the redundant packets is critical as they inject more traffic load and potentially cause network congestion. Since video frames are generated in fixed rate, video frames are separated by fixed time distance; IFG (see Chapter 5).

In order to avoid the redundant packets competing with the original packets, the effective transmission time of the redundant packets (T_{Fir}) is between frames F_i and F_{i+1}:

$$T_{Fir}=T_{Fi} +(T_{Fi+1} - T_{Fi})/2=T_{Fi} +(IFG)/2 \quad (6.2)$$

6.2.4 Performance evaluation

Radio propagation is affected by path loss, shadowing and multipath transmission. Consequently, the channel produces signal degradation and packet errors. In previous study, the channel was considered a line of sight and modelled as two-ray ground propagation. This was taken as video surveillance is usually implemented in a well-known environment, where proper adjustment is made during initial installation. Consequently, the proposed methods perform in the best case scenario.

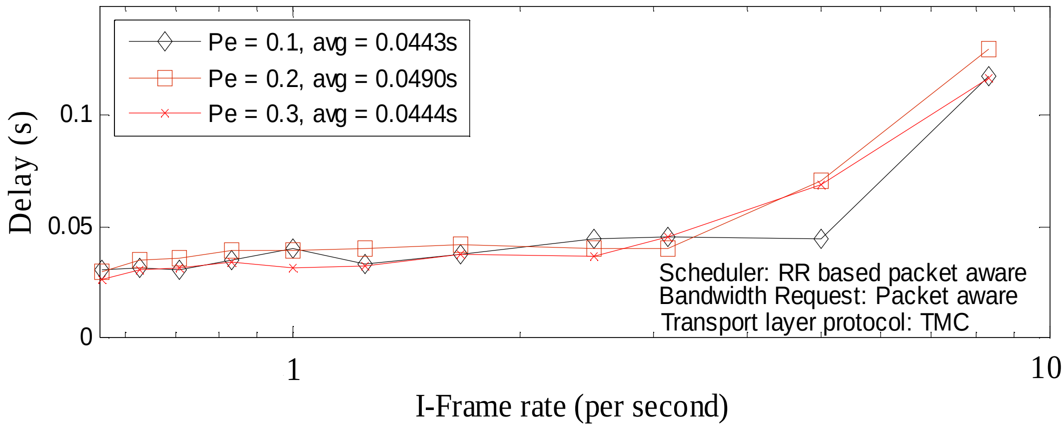
In this study, the non-line of sight condition and interference are considered. The erroneous circumstances are presented by inserting channel and error models in WiMAX simulation environments. The channel and error models have been discussed in Section 2.7.3 and Section 2.7.4. The simulation scenario follows Section 2.7.5.

6.2.5 Performance degradation

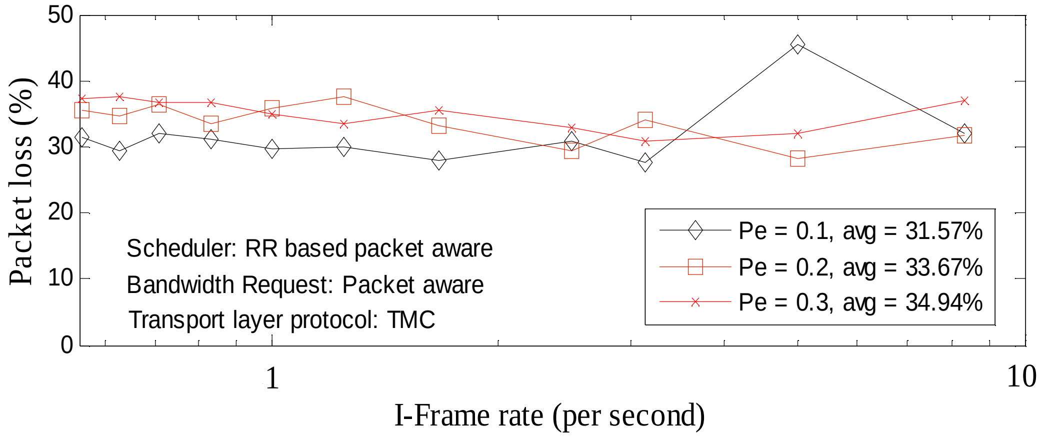
Previous proposed methods improve the surveillance performance in three stages, bandwidth request mechanism, scheduler algorithms and transport layer protocol. Bandwidth request is packet-aware which implements contention request for I-frames and next piggybacking for P-frames, non-sorting scheduler algorithms based on either

RR or FIFO, and transport layer protocol uses IR protocol assisted by MAC support (TMC protocol).

Figure 6.2 shows packet delay and packet loss performances of WiMAX-based video surveillance when the channel is erroneous. The simulated WiMAX uses the proposed RR-based packet-aware non-sorting scheduler, packet-aware bandwidth request mechanism and TMC protocol. The error model was set for the probability of error, P_E, 0.1, 0.2 and 0.3.



a. Packet delay



b. Packet loss

Figure 6.2: Video surveillance performances in error-prone environment

Packet delays are, on average, relatively consistent with channel error rates, fluctuating slightly around 0.046s for the three P_E values. In contrast, packet loss increases as channel error rate increases. This shows that an error-prone environment introduces more packet loss problems than packet delays. Therefore, the application layer techniques focus on reducing losses in order to improve the received video quality, rather than decreasing transmission delay.

6.2.6 Redundancy performance

The erasure channel in wireless transmission increases instances of packet loss. The receiver generates more packet retransmission, but the NACK packets can also be lost during transmission. As a result, less data are received and video quality is reduced. Additional redundancy injects a new traffic and increases queues. The longer queue makes the packets experience more delay. As shown in Figure 6.3, both TMC protocol with transport-layer and rateless code redundancies cause greater delays than only TMC protocol. Significant delay increment occurs when I-frame rates are high. The higher rate I-frame rates, the higher redundant bytes.

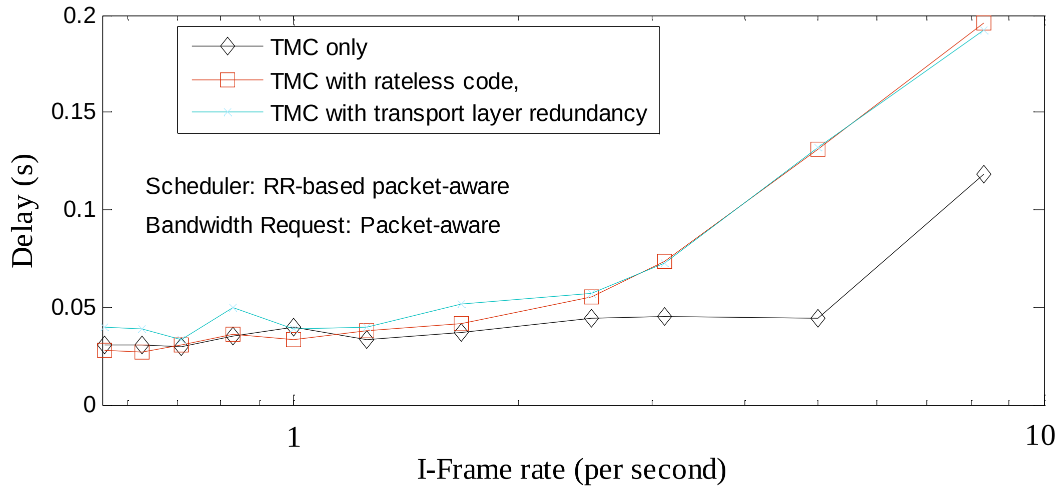


Figure 6.3: Delay in error-prone environment

Redundancy in the transport layer causes the sender and receiver to make more effort in relation to stamping and retransmission. Consequently, the delay increases more than non-transport-layer redundancy. On average, transport-layer redundancy produces a 29 ms delay increment, while rateless code is only 24 ms.

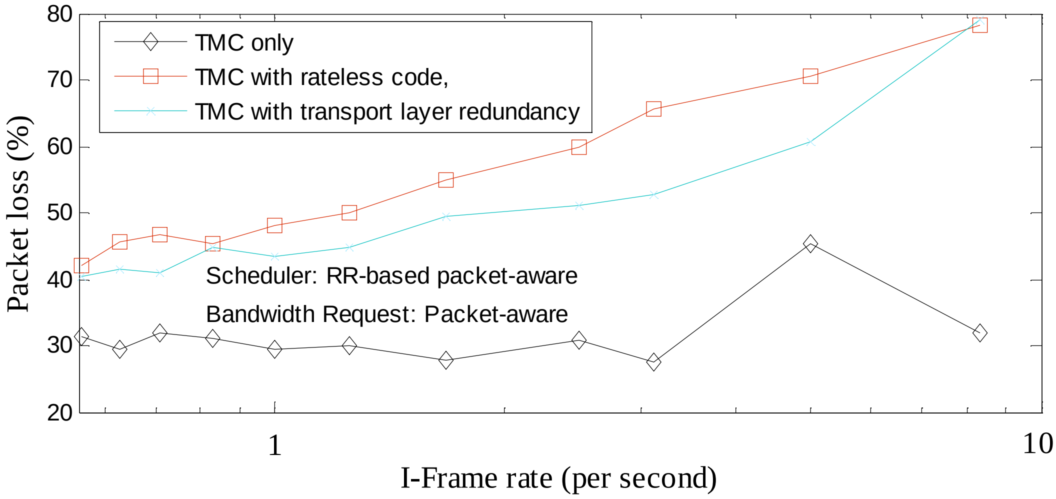


Figure 6.4: Packet loss in error-prone environment

Since rateless code redundancy is rendered useless when the amount of received data (m) is less than the original one (k), its packet loss is higher than transport-layer redundancy. Figure 6.4 shows that transport-layer redundancy has 15.87% lower packet loss than rateless code redundancy. However, rateless code produces higher PSNR increments of up to 3.61dB (Figure 6.5). Transport-layer redundancy failed to improve video performances. Overall, rateless code improves the UL video streaming performance by about 1.34 dB on average.

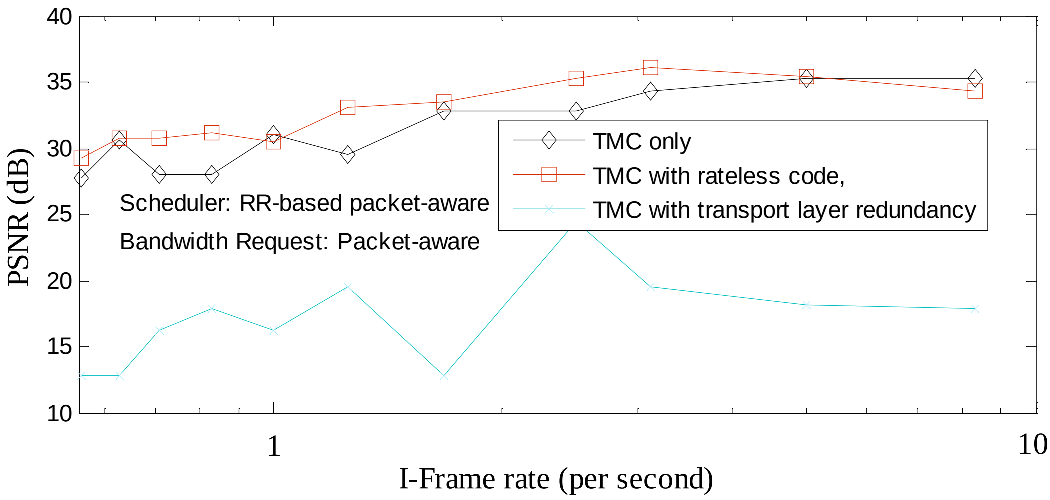
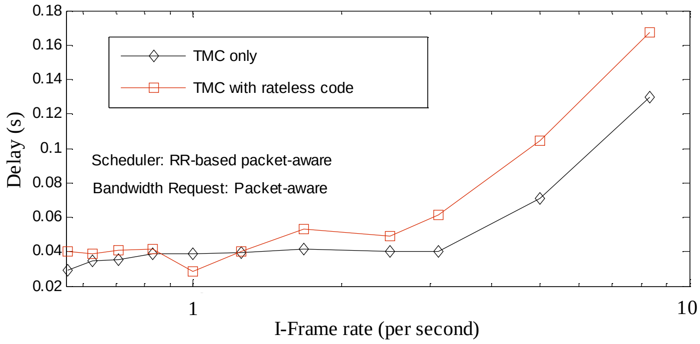
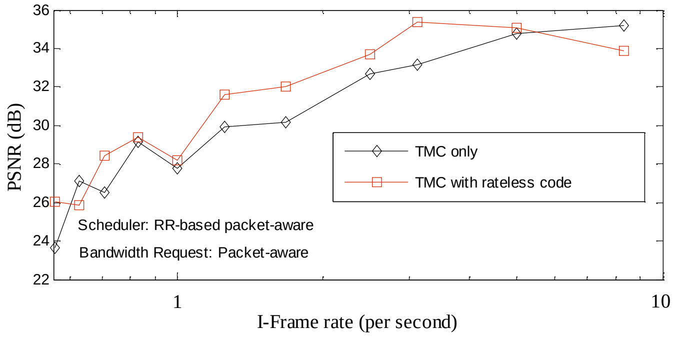


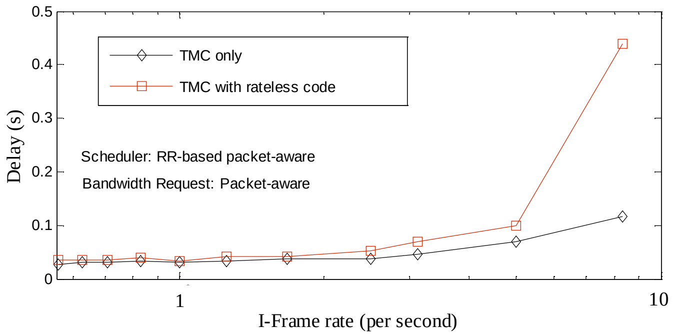
Figure 6.5: PSNR in error-prone environment



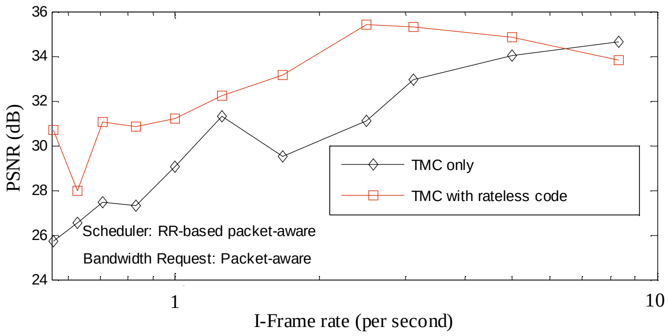
a. Delay for $P_E = 0.2$



b. PSNR for $P_E = 0.2$



c. Delay for $P_E = 0.3$



b.PSNR for $P_E = 0.3$

Figure 6.6: Rateless code performance for various P_E

Figure 6.6 demonstrates the consistency of the rateless code when the wireless channel condition deteriorates. By increasing P_E to 0.2 and 0.3, the rateless code is still able to improve the received video quality.

6.3 Post-decoding error concealment techniques

Although efforts to recover packet loss have been performed in MAC, transport and application layers, loss may still exist. Error concealment is defined as the last means to overcome it within the application layer. There are two known methods of error concealment: frame copy (FC) and motion vector (MV).

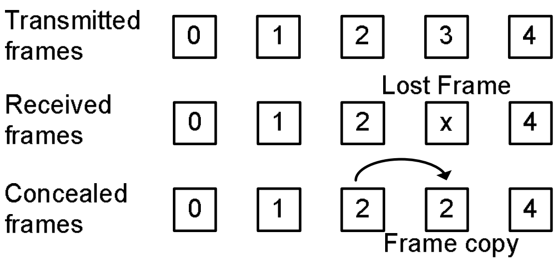


Figure 6.7: Frame copy error concealment

Most researchers focus on MV-based error concealment methods, while the FC method illustrated in Figure 6.7 is considered too simple, as it produces completely different images. However, the FC method in post-decoding error concealment techniques can be used as an additional improvement.

This section explores some post-decoding FC error concealment methods that apply after decoding process to replace dropped frames. The methods may be applied to conceal a whole lost frame or a partial error frame.

6.3.1 Inline FC method

Inline FC error concealment method is intended for video transmission that has periodical images, such as a static surveillance camera or fixed periodical moving camera with period of T-frames. Inline FC uses the co-located previous frame as the source of improvement. One full period of previous images is stored in buffer to be used for error concealment. Algorithm 1 shows how the method works.

Algorithm 1: Inline frame copy

if frame n (F_n) is missing, **then**

Determine previous frame index: m $\leftarrow$n-1

Determine period position: x $\leftarrow$ n/T

Obtain the frame copy source: $FC_{(m-xT)}$

Conceal the lost frame: $E(F_n) \leftarrow FC_{(m-xT)}$

end if

6.3.2 Closed related FC method

This method replaces the dropped frame by using the most similar neighbouring frame. This is classified as a pre-transmission error concealment technique, whereby the closest frame is analyzed prior to transmission by comparing the distance of current frame with the previous and subsequent frames. The closest related frame index is then stored within the previous frame's header. This index will be used to determine which packet can be a frame copy source if the current frame is missing. Algorithm 2 depicts the error concealment method.

Algorithm 2: Closed related frame copy

Sender

Prepare sending frame n-1

if $\delta(F_n, F_{(n-1)})$ is less than $\delta(F_n, F_{(n+1)})$ **then**

Put x in header frame n-1: x $\leftarrow$ 0

else

Put x in header frame n-1: x $\leftarrow$ 1

end if

Send frame n-1

Receiver

if frame n (F_n) is missing, **then**

Determine previous frame index: m $\leftarrow$n-1

Obtain index in previous frame header: x $\leftarrow$ F_m header

if x$\leftarrow$0

Estimate lost frame using previous frame copy:
$E(F_n) \leftarrow FC_{(n-1)}$

else

Estimate lost frame using next frame copy:

$E(F_n) \leftarrow FC_{(n+1)}$

end if

end if

6.3.3 Referenced partial FC method

The referenced partial FC is a pre-transmission error concealment technique that requires frame analysis prior to transmission. As shown in Algorithm 3, frame n (F_n) is divided into smaller blocks with index i, B_n^i where $F_n = \sum_{i=0}^{A_n/A_B} B_n^i$. The number of blocks B_n^i depends on block size A_B and frame size A_n. Each block is then compared with other blocks from neighboring frames (frame n-1 and n+1). By comparing distance $\delta\left(B_{n-1}^j, B_n^i\right)$ to $\delta\left(B_{n+1}^j, B_n^i\right)$ $where\ i = j$, a value of the overhead bytes in position i is obtained. Overhead values are stored in frame n-1 header. At the receiving end, the lost frame will be estimated using $E(F_n) = \sum_{m,j} BC_m^j$.

Algorithm 3: Referenced Partial Frame Copy

Sender

Initiate number of partial frames (blocks)

i ← frame size/block size

for all partial frames (blocks, B^i)

if $\delta(B^i_n, B^i_{(n-1)})$ less than $\delta(B^i_n, B^i_{(n+1)})$ **then**

bytes header frame n-1

h(i) ← 0

else

header frame n-1

h(i) ← 1

end if

end for

Send frame n-1

Receiver

if frame n (F_n) is missing, **then**

Determine previous frame index and fetch frame header

m ←n-1

h(i) ← F_m header

for index 0 to i

if h(i)==0

Estimate block using previous frame's block copy

$E(B^i_n) \leftarrow BC^i_{(n-1)}$

Else

Estimate block using next frame's block copy

$E(B^i_n) \leftarrow BC^i_{(n+1)}$

end for

end if

6.3.4 Combined partial FC method

Combined partial FC method overcomes the drawback of the referenced partial FC, which is an overwhelming error concealment header when block size is very small. The combined partial FC does not require frame processing prior to transmission.

As shown in Algorithm 4, the method obtains the estimated blocks $E(B_n^i)$ by combining 2 partial frame copies of BC_{n-1}^i and BC_{n+1}^iconsecutively. Each neighboring frame will contribute to 50% of all estimated frames $E(F_n)$.

Algorithm 4: Combined Partial Frame Copy

Receiver

if frame n (F_n) is missing, **then**

Initiate combination

flip ← true

for all partial frames

if flip==true

Estimate block using previous frame's block copy

$E(B^i_n) \leftarrow BC^i_{(n-1)}$

flip ← false

Else

Estimate block using next frame's block copy

$E(B^i_n) \leftarrow BC^i_{(n+1)}$

flip ← true

end for

end if

6.3.5 Pattern partial FC method

This method uses two previous frames and one subsequent frame. The patterns are identified for both previous and subsequent frame patterns. If frame n is the missing frame, the previous pattern is represented by measuring the distance between frames n-1 and n-3. The next frame pattern is obtained from frame n-1 and n+1. The closest distance determines whether frame n-1 or n+1 is to be used to estimate missing frame. The pattern partial FC method is outlined in Algorithm 5.

Algorithm 5: Pattern Partial Frame Copy

Receiver

if frame n (F_n) is missing, **then**

for all partial frames

if $\delta(B^i_{n-3}, B^i_{n-1})$ less than $\delta(B^i_{n-1}, B^i_{n+1})$

Estimate block using previous frame's block copy

$E(B^i_n) \leftarrow BC^i_{(n-1)}$

Else

Estimate block using next frame's block copy

$E(B^i_n) \leftarrow BC^i_{(n+1)}$

end if

end for

end if

6.3.6 Evaluation method and results

The proposed error concealment techniques are evaluated using the JM15.0 H.264/AVC Reference Software [133]. Video sequences "mobile_qcif.yuv" and "mobile_cif.yuv" are selected to evaluate the proposed method performances. The QCIF sequence has resolution of 176x144 and the CIF sequence resolution is 352x288. The details of the simulation parameters are presented in Table 6.1.

Table 6.1: Simulation parameters

Parameters	Values
Frame rate	30 fps
Frame type	IPP
I-Frame period	15
Number of referenced frame	1
Quantization Parameter	22, 24, 30 and 36

Each frame is encoded into a single slice, which is transported within one packet. In order to simulate dropped packets, one is dropped in every GOP. The dropped frames are then concealed using the proposed methods, which are variants of FC. The results are then compared with the original FC method.

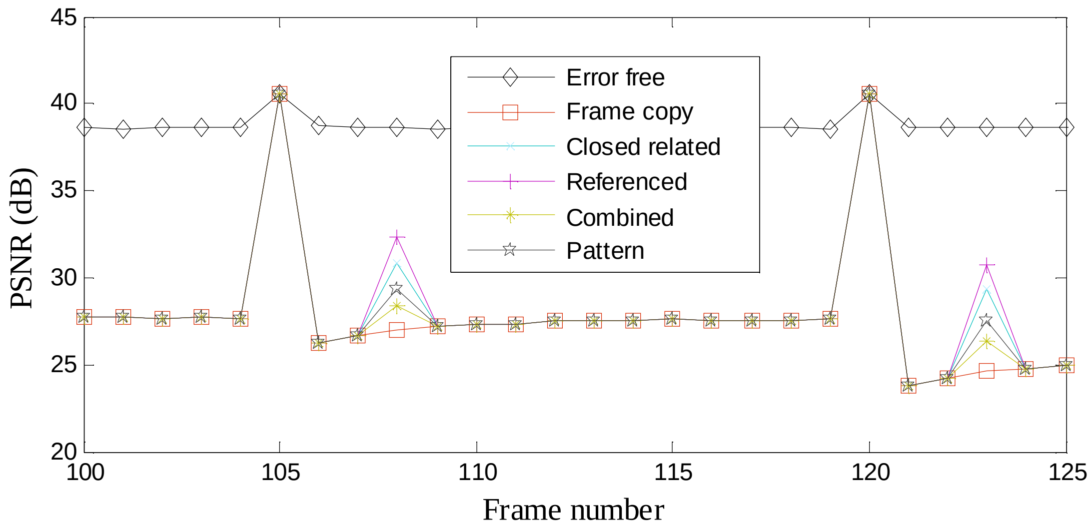


Figure 6.8: Post-decoding FC performance

Figure 6.8 shows the error concealment performances for QCIF image using the PSNR parameter. The proposed methods are able to increase individual lost frames in various degrees. The referenced partial FC increases significantly the quality of the concealed frame. All other methods perform better than the existing FC method.

Table 6.2 presents the average PSNR improvement over erroneous frames. The closest related FC method is able to increase the average performance by up to 2.27dB. The referenced partial FC method outperforms significantly, increasing the PSNR up to 4.98 dB (QCIF) and 6.64 dB (CIF). The combined and pattern methods yield slighter improvement than the referenced method.

Table 6.2: The average PSNR over erroneous frames

Seq	QP	Error Free	FC	CR FC	Ref. PFC	Combined PFC	Pattern PFC
QCIF	22	38.69	26.45	28.29	31.43	27.51	28.38
	24	36.76	26.36	28.25	31.30	27.40	28.31
	30	30.88	25.52	26.29	26.54	26.54	27.23
	36	27.08	24.29	25.26	27.61	25.00	25.43
CIF	22	39.2	21.27	23.77	27.87	22.82	23.71
	24	37.36	21.25	23.78	27.89	22.92	23.76
	30	32.39	21.14	23.65	27.75	22.72	23.58
	36	27.78	20.82	23.09	26.82	22.32	23.08

In terms of the PSNR improvement, those frame copy based methods are comparable to MV-based methods. According to Bo and Gharavi [124], the Hybrid Motion Vector Estimation (HMVE) method is able to improve PSNR over the FC method up to 4.79 dB for QCIF sequence and 4.32 dB for CIF sequence, while the pixel-based MV extrapolation (PMVE) method reaches improvement levels of 3.70 dB and 3.52 dB.Furthermore, Tai et al. [123] claimed that the object-based full frame error concealment (OFFC) method is able to increase the PSNR up to 3.123 dB, 2.614, and 1.981 dB for mobile CIF sequence with QPs 24, 30, and 36.

These facts prove that the proposed FC-based methods are able to gain quality as MV methods do, although they work only on the lost frame.

6.4 Summary

This chapter implemented and proposed passive streaming techniques in the application layer: redundancy and error concealment techniques. The implementation of the rateless code redundancy shows the consistent improvement on the quality of the received video. The many-to-many redundancy relation in rateless code is proven to be much more effective than one-to-one redundancy in relation to transport layer redundancy. The best time to transmit the redundancy packet is in between the original frames.

This chapter also proposed some post-decoding error concealment methods to reconstruct the dropped frames. The simulation has highlighted the positive outcome of the proposed methods by improving PSNR up to 6.64 dB over the FC method. Furthermore, these methods improve only the PSNR of lost frames; the PSNR performance improvements are comparable with MV-based methods.

Chapter 7
Mathematical model and analysis

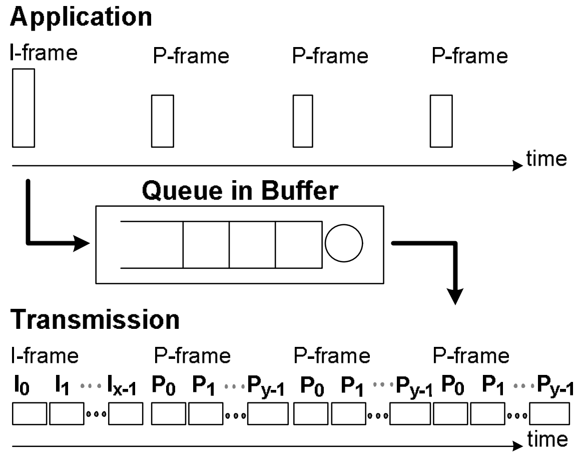
7.1 Introduction

This chapter presents a mathematical model for NACK-based transport layer protocol analysis in 802.16 environment and presents the assumptions that have been made. Initially, mathematical models are made separately between the application layer, transport layer and MAC layer. Finally, they are combined to characterize the performance of the NACK-based transport layer protocols in terms of packet loss and delay.

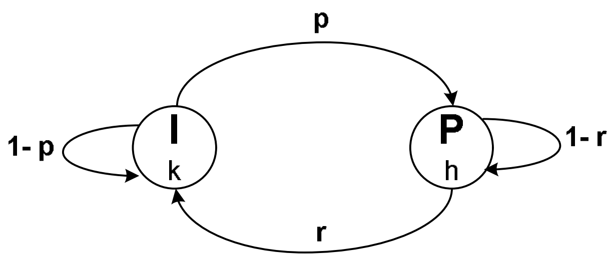
The network analyzed in this chapter is assumed to be in point-to-multipoint mode, where the video streaming application is in UL direction (many-to-one) as the performance is more critical than that of a broadcasting direction (one-to-many). It is assumed that all mobile nodes have similar video coding parameters, such as the sequence of I and P frames (IPP sequence), and the number of I and P frame grouping (GOP). This assumption is made as this thesis examines separately the effect of video rates. This thesis also assumes that mobile nodes have a sufficient buffer to ensure no packet loss is caused by overflow.

7.2 Application layer model

Unlike the research in [135], which considered that the application sends a video frame immediately, the model approaches the real situation where frames are fragmented and sent in several packets, as shown in Figure 7.1a. I-frame is fragmented into x packets, and P-frame into y packets. There are two transmission states; sending I or P-frames. By making this assumption, video transmission can be modelled using a two-state Markov chain (Figure 7.1b). Modelling the MPEG4 stream with IPP frame sequence using a two-state Markov chain model is also performed by [135]. However, the proposed analysis combines link and upper layers.



(a). Video generation and transmission



(b). Video transmission model

Figure 7.1: Application layer model

Since video contains I and P-frames, there are two transmission states; sending I-frame packets (I packets) and P-frame packets (P packets). The probability of sending I and P packets are denoted respectively as k and h. Besides the transmission states, the application layer has an idle state where no packet is sent. However, packets sent by application layer are stored in the MAC layer buffer before transmission, which means that packets remain in sending node during idle state in application layer. Therefore, the idle state is skipped of the model. Consequently, probability k is the percentage of I packets within 1 GOP and probability h is the percentage of P packets within 1 GOP. Variables p and r are the probability of sending P packets after sending I packets, and vice versa. The probability of sending sequential packets is denoted as $(1 - p)$ and $(1 - r)$.

7.3 Transport layer model

Since the traffic sources are known, each probability variable is determined by the transport protocol model presented in Figure 7.2. For the UDP model, from x sequence of I packets, only one packet is followed by P packet. Likewise, after sending y sequence of P packets, only 1 is followed by I packets. Therefore, $p = {}^{1}/_{x}$ and $r = {}^{1}/_{y}$. The stationary state probabilities of the two-state Markov chain model in Figure 7.1b are expressed as:

$$\pi_I = \frac{r}{p+r} \tag{7.1}$$

$$\pi_P = \frac{p}{p+r} \tag{7.2}$$

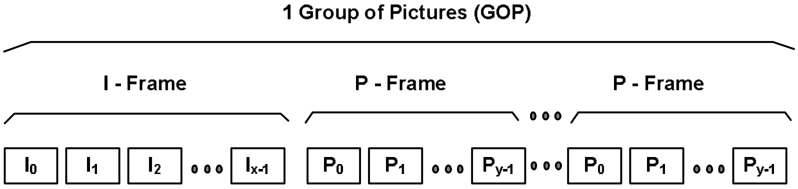
The probability of error (P_E) is calculated according on the fact that an error occurs when a receiver receives a packet that is neither I packet nor P packet. P_E is given by:

$$P_E = (1 - k - h) * (\pi_I + \pi_P) \tag{7.3}$$

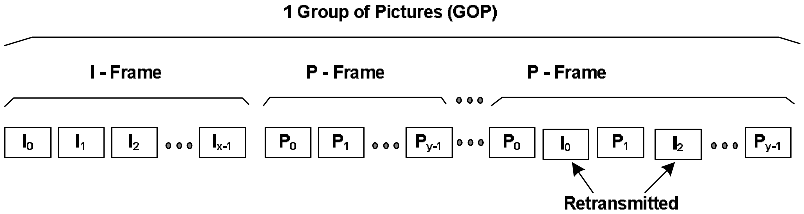
As shown in Equation 7.3, when the channel is lossless, $P_E = 0$ because $k + h = 1$.

In UDP, the packet arrival rate in transmission (λ_t) is equal to that of the traffic source (λ_s). Both QR and IR scheduling experience rate increments as the senders should also retransmit the lost packet(s). Since retransmission is applied only to prioritized packets, the retransmission rate (ret) is determined by the probability of error of I packet.

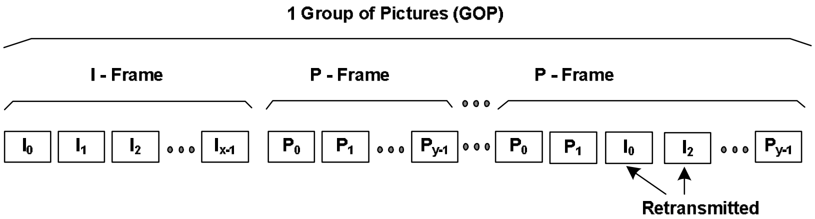
Let δ be the loss rate of I packets, then the retransmitted packet is $\delta.x$ where x is the average number of packets of I frames, as demonstrated in Figure 7.2. The loss rate (δ) is taken from the UDP probability of error. The number of the NACK packets in QR is equal to or θ times higher than in IR as its receiver may generate more than one NACK for multiple lost packets in one frame. Each NACK interruption results in an additional load for the receiver and sender. If the additional load is assumed to be proportional to the retransmitted content, the retransmission rate is tripled ($3\delta x$). The total arrival rate, $\lambda_t = \lambda_s + ret$, where $ret = (1 + \theta)3\delta x$. $\theta = 0$ for inter-frame and $\theta \geq 0$ for quick response.



(a). UDP model



(b). QR NACK (BVS protocol) model



(c). IR NACK (IR protocol) model

Figure 7.2: Transport layer protocol model

Since I frame size (L_I) is often much higher than P frame size (L_P), the arrival rates of traffic source (λ_s) for both frame types are different. In order to obtain a more precise result, arrival rates were calculated for both packet types separately. The arrival rates depend on the maximum transfer size ($byte_{MAX}$) of the WiMAX network for each SS. Let the frame rates of the source be fps, the arrival rates of I packet (λ_I) is given by:

$$\lambda_I = \frac{\left(\frac{fps(L_I + ret_{REQ})}{GOP}\right)}{byte_{MAX}} \qquad 7.4$$

The arrival rate for P packets (λ_P) is:

$$\lambda_P = \frac{fps\left(1 - \frac{1}{GOP}\right)L_P}{byte_{MAX}} \qquad 7.5$$

The value of retransmission request (ret_{REQ}) depends on the type of the transport layer protocols. For UDP, $ret_{REQ} = 0$ as UDP does not retransmit the lost packets. IR requires maximum one additional request for the retransmitted packet as it retransmits lost packet in a frame at once; $ret_{REQ} = 1$. BVS protocol requires at least one request for the retransmitted packets if NACK packets for the same frame arrive at the same WiMAX frame. Otherwise, multiple NACKs for multiple losses in a frame result different requests. The BVS $ret_{REQ=}1 + \varphi$, where φ is the request increment caused by multiple NACKs. It is assumed that $\varphi = 0.001$, which means only 1 out of 1000 multiple NACKs produces multiple requests. This assumption is considered as DL traffic consists largely of NACK packets. The φ value increases significantly when DL traffic is busy.

TMC protocol uses IR in the upper layer. However, since bandwidth for the retransmitted packet is appended to the nearest request, the arrival rate for the retransmitted packets is 0 ($ret_{REQ} = 0$).

7.4 WiMAX network model

The WiMAX frame provides K-slots for bandwidth requests. The SSs contend in order to obtain bandwidth allocation. As shown in Figure 7.3, contention slots are located in the UL-sub-frame. BS informs SSs the successful request through the UL-MAP in the DL-sub-frame. Then, the SS with successful request sends its data in the respected UL burst of the UL-sub-frame.

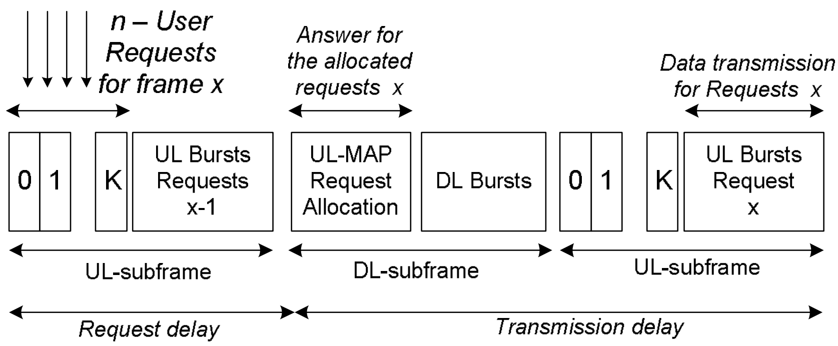


Figure 7.3: Request slots and their allocation

A successful request determines the transmission's success. Contention request analysis provided by [62] demonstrates that the probability of one transmission experiences collision is:

$$p_{collision} = 1 - (1 - \pi)^{n-1} \tag{7.6}$$

The probability that a mobile node transmits successfully a packet is expressed as [63]:

$$p_{succ} = \frac{n\pi(1-\pi)^{n-1}}{1-(1-\pi)^{n}} \tag{7.7}$$

The probability reduction caused by the increase of number of mobile nodes is considered. The greater the mobile node number, the lower the transmission probability. If the arrival rate (λ_t) of each mobile node in random access environment is known, then the probability π that a node transmits a packet can be expressed as $\pi = \lambda_t/n$, where n is the number of mobile nodes. The equation has been used by Q. Ni et al. [63] for random access in WiMAX environment analysis. Equation 7.7 can be rewritten as:

$$p_{succ} = \frac{\lambda_t(1-\lambda_t/n)^{n-1}}{1-(1-\lambda_t/n)^{n}} \tag{7.8}$$

7.5 Packet loss

If every error packet is considered as packet loss, then the total packet loss can be projected using Equations 7.3 and 7.8. The probability of error (P_E) becomes:

$$P_E = \left(1 - k.\,p_{succ(I)} - h.\,p_{succ(P)}\right) * (\pi_I + \pi_P) \tag{7.9}$$

Packet loss can be calculated as:

$$Loss = (fps/GOP).\,x.\,(1+\delta).\,P_{E(I)} + (fps - fps/GOP).\,y.\,P_{E(P)} \tag{7.10}$$

The first component is I packet loss and the latter is P packet loss.$P_{E(I)}$ and $P_{E(P)}$are the probabilities of error based on arrival rates of I and P packets.

7.6 Packet delay

There are two main delay components: the delay that a packet suffers on the buffer of the mobile node, and medium access delay. Since buffer in SS is assumed to be large enough to store the generated video, simple M/M/1 system is used to calculate delay in SS. The waiting time (W_T) is given by:

$$W_T = \frac{\lambda/\mu}{\mu-\lambda} \tag{7.11}$$

The service rate (μ) is determined by number of maximum requests allocated by network.

As shown in Figure 7.3, the access delay contains the request delay (d) and transmission delay (tx_{TMC}). The request delay is computed based on the probability of the successful request. When the number of the SSs is higher than the contention slots, delay is given by [63]:

$$d = \frac{(n/K)fd}{(1-1/n)^{n-1}}, n = iK, i = 1,2,.. \quad (7.12)$$

Otherwise,

$$d = \frac{fd}{(1-1/K)^{n-1}}, n < K \quad (7.13)$$

Constant fd is WiMAX frame duration. For UDP, BVS and IR, the transmission delay (tx) is equal to one WiMAX frame duration. For TMC protocol, when receiving NACK packet, MAC in SS requests bandwidth even though regular data are not available. Consequently, there is a delay reduction. The TMC transmission delay is given by:

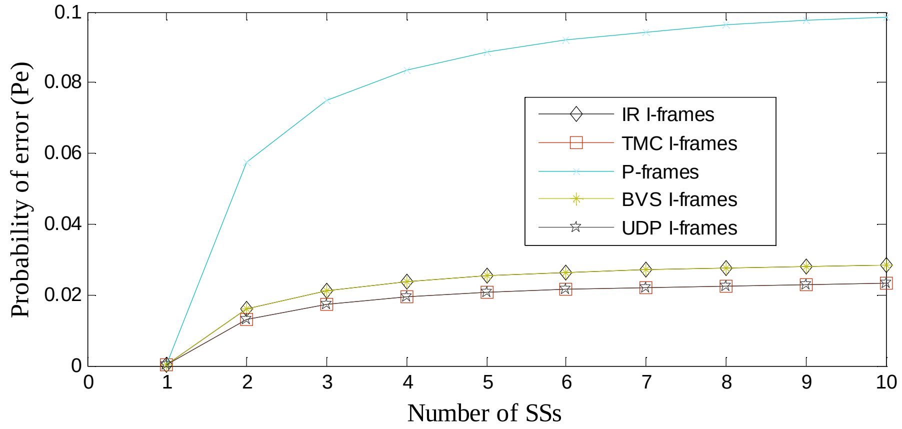
$$tx_{TMC} = fd.\left(fps - \frac{fps}{GOP}\right) \quad (7.14)$$

7.7 Results of the analysis

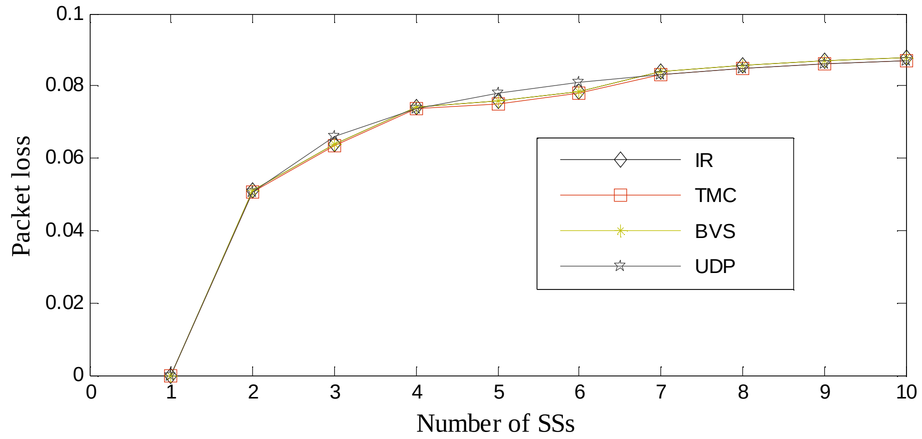
In order to analyze mathematically the proposed protocols, the assumptions were made for some variables. The application layer and MAC layer variables were obtained from previous simulations. The number of contention slots (K) is set to one, with frame duration of 4ms. First, only the case where GOP = 30 is considered; then, the number of SSs is varied from 1 to 10 and the performance parameters are plotted in Figure 7.4.

The number of SSs was chosen based on the fact that with $K = 1$, the maximum arrival rate should be less than $\frac{K}{frame\_duration}$ to avoid frequent collision. This selection is sufficient to compare the performance of the proposed protocol.

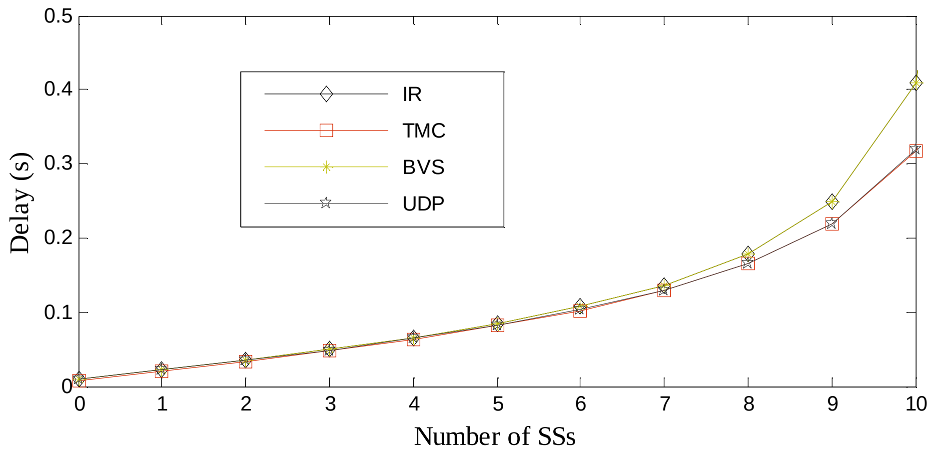
The behavior of the compared protocols for different number of SSs is given in Figure 7.4. Since UDP transmits only the original load, the protocol experiences a low P_E. As expected, the proposed TMC protocol also has a low P_E as the ret bytes are transmitted in additional bandwidth. BVS and IR experience higher P_E as both protocols retransmit the lost packet(s) without additional bandwidth. As the number of SS increases, so does the error probability.



a. Probability of error



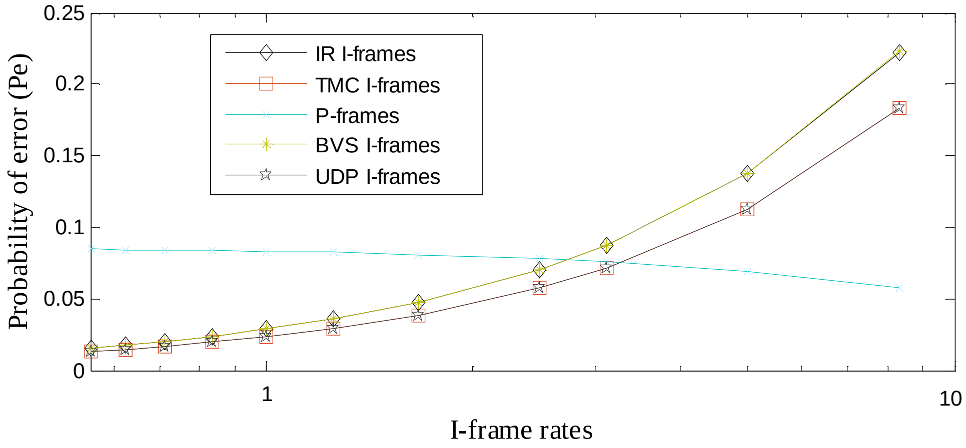
b. Packet loss



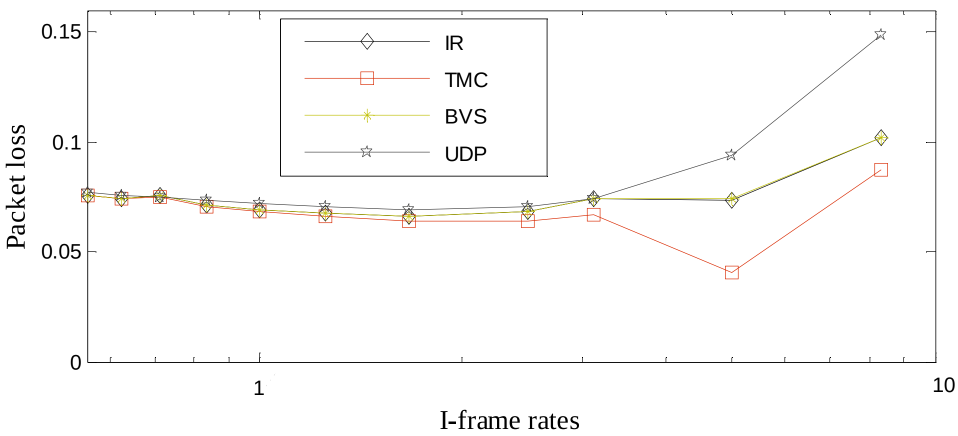
c. Delay

Figure 7.4: Protocol performances for various numbers of SSs

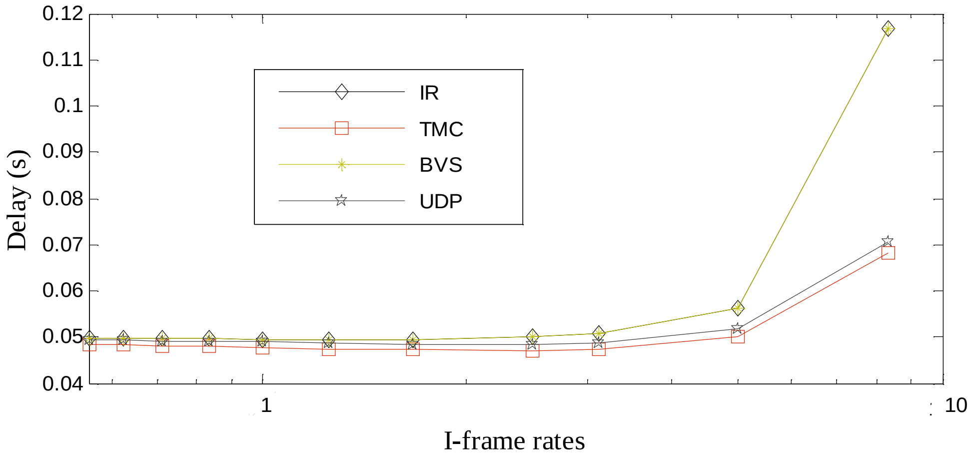
All protocols have the same P_E for P packets as no retransmission applied to these packets. The P_E of P packets is higher than I packets because when using GOP=30, the average number of P packets (87 packets) is higher than of I packets (16 packets).



a. Probability of error



b. Packet loss



c. Delay

Figure 7.5: Transport layer protocol performances

Packet loss and delay increase when SSs transmit more bandwidth requests. BVS yields the higher packet loss as its bandwidth request increases by addition $1 + \varphi$, followed by IR by factor 1. The TMC protocol produces the lowest packet loss and delay as TMC experience similar number of request as UDP and transmission delay reduces by factor $\mathrm{fps} - \frac{\mathrm{fps}}{\mathrm{GOP}}$. This behavior is desired as the aim of the proposed TMC protocol is to reduce delay and to increase video quality (packet loss reduction).

Figure 7.5 plots the behavior protocols for various I-frame rates. The higher I-frame rates, the more frequent I-frame packets. $P_E$of I-frame packets increase whenI-frame rate increases. In contrast, P_E of the P packets increase when I-frame rate decreases. Again, the proposed TMC protocol outperforms other protocols.

7.8 Summary

In this chapter, the application layer and the NACK-based protocols in the WiMAX environment were modeled. The application layer is modeled as the sequence of video packets with two conditions, sending I-frames' packets and P-frames' packets. The idle condition is ignored as packets from upper layer are hold in MAC buffer until data transmission. The NACK-based protocol is modeled with the retransmitted packets included. From those model, the steady state probabilities are then determined. Then, the probability values are inserted to the existing WiMAX model, where traffic sources are calculated based on the type of transport layer protocols.

Delay comprises two parts: delay in buffer and delay in transmission. The buffer delay of each protocol is determined by the traffic rates. The transmission delay is determined by the number of bandwidth requests for each protocol.

From the proposed model, packet loss is calculated from the probability of error. Consequently, packet loss and delay characteristics of the assessed protocols were obtained. GOP 30 was used as a sample to determine the impact of increasing number of SSs. Packet loss and delay increase exponentially when the number of SS increases. The P-frames experience the highest error probability as the number of P-frames is greater than the number of I-frames.

For various I-frame rates, P-frames of each assessed protocols experience similar probability of error and packet loss rate. This is caused by the retransmission being applied only for I-frames, which contribute largely to the total packet loss for each protocol.

UDP experiences the most packet loss compared with other protocols as UDP does not make sufficient effort to avoid losses. TMC has the lowest packet loss as it not only retransmits the lost packets, but also utilizes the higher bandwidth.

TMC protocol is superior in terms of delay, followed by UDP. TMC has lower delay than UDP as TMC utilizes more bandwidth. IR and BVS delays experience an almost identical delay.

Chapter 8
Experimental performance evaluation

8.1 Introduction

The objective of the chapter is to evaluate the feasibility of implementing the proposed method in a real network. Although each method assessment may experience different channel conditions over time, repeating the experiments may be adequate to represent the sample of network implementation. The experimental evaluations clarify the expectations and limitations offered by the WiMAX system, as well as the performance of the proposed methods.

WiMAX experimental research has been conducted by academic researchers. Scalabrino [1] pioneered this using a real WiMAX network testbed with which to measure channel performance. Ruiz et al. [136] evaluated video performance within a fixed WiMAX connection and improved the quality of the received video by using the automatic video coding rate adjustment (AVCRA). Furthermore, Pentikousis et al. [137] investigated VoIP and video streaming on WiMAX systems with the Darwin Streaming Server in BS side; accessed by 2 SSs. Issa [138] used an attenuator to simulate channel losses between WiMAX devices. Some researchers combined WiMAX networks with existing networks, such as in [139]. Halepovic [139] experimented with Skype and real player applications on WiMAX. The observed server was connected to the WiMAX BS testbed via an internet connection that may contain other networks.

This chapter presents an experimental evaluation for the proposed transport layer protocol and the application layer technique. Despite the effort to implement the MAC layer improvement in the experimental work, the proposed MAC methods outlined in Chapter 4, along with the transport-MAC cross-layer protocol, cannot be realized as access to the MAC of the existing WiMAX device is restricted. Transport layer protocol evaluation involves only video packet transmission, while the application layer technique involves video coding; therefore, video streamers for both layers are designed separately.

8.2 Experimental network

The tests were performed using the experimental WiMAX network illustrated in Figure 8.1, which contains 1 BS Aperto PM-3000 with two SSs PacketMax-120. The monitoring unit or video receiver PC is attached to the BS.

The PacketMAX 3000 BS is part of the PacketMAX family produced by Aperto Networks [140] and the WiMAX device is certified by the WiMAX Forum. PacketMAX 3000 claims to be an economic solution using a physical space-saving design that is able to serve premium voice, multimedia and data services. The BS can operate as a single sector BS or as an operation within a multi-sector cell site. It supports fixed and nomadic WiMAX applications. PacketMAX 3000 operates in licensed and license-exempt bands under virtually all wireless conditions, including line of sight (LOS), obstructed LOS and non-LOS. Moreover, PacketMAX 3000 can serve as a wireless access point for hundreds of simultaneously active subscriber units, serving both outdoor and indoor applications.

PacketMAX 120 is part of Aperto's PacketMAX 100 series subscriber units; it is compliant with IEEE 802.16-2004 standard [140]. PacketMAX 120 operates in 2 GHz, 3 GHz, and 5 GHz bands using time division duplex (TDD). PacketMAX 120 can be deployed for both licensed and license-exempt applications, and it is an outdoor radio equipped with an integrated or optional external antenna for greater gain.

In the experimental network, Aperto’s PacketMax-3000 uses a wireless sub-system (WSS); a BS radio PM-BSR-58 that operates within the spectrum of 5.8 GHz (5.725-5.925GHz). The system has a channel width of 3 MHz to 7 MHz. The experiment configuration used 3.5 MHz channel width, UL and DL modulations are QPSK, and transmit power was 5 dBm. The WiMAX service used the best-effort service and the cyclic prefix was set to 1/16.

The SSs used the internal antenna and was pointed towards the BS. The SS1 was located about 5 m from BS with a Constant Bit Rate (CBR) video streaming to the monitoring unit in BS. The SS2 was located about 20 m from BS, streaming the evaluated video source.

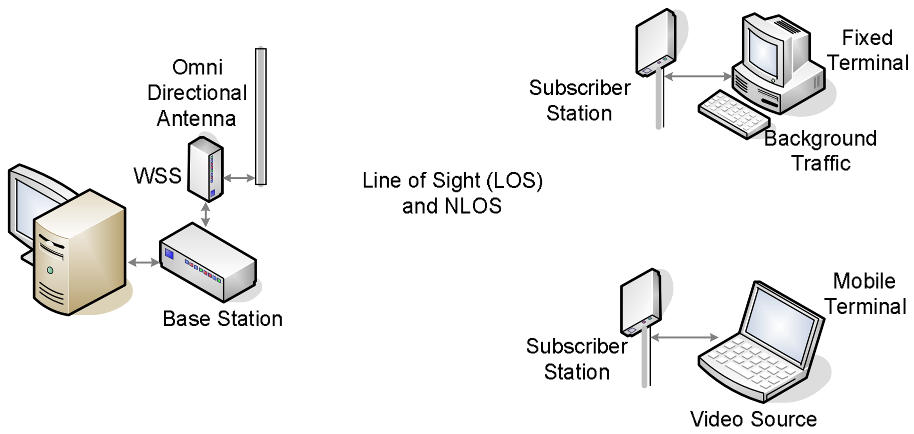


Figure 8.1: Transport layer protocol performances

8.3 Transport layer protocol

This thesis evaluates the proposed IR protocol and compared it with BVS and UDP. The proposed protocol and BVS are implemented by modifying the UDP packet and using two-way UDP connections: one for data stream and another for NACK transmission.

8.3.1 Video streamer

Video streamer was implemented using Java programming language. The NACK-based protocols are realized in the existing UDP protocol by adding additional 32 bytes header to the data part of the UDP protocol. To stream video data equally, zero bytes were added to UDP data. The packet structure is shown in Figure 8.2.

Source IP address	Destination IP address	Zero, Protocol, UDP length	Source & Dest. Port	Data length & checksum	1024 bytes data+32 bytes zeros

(a) UDP packet format

Source IP address	Destination IP address	Zero, Protocol, UDP length	Source & Dest. Port	Data length & checksum	BVS header	1024 bytes data

(b) Quick response packet format

Source IP address	Destination IP address	Zero, Protocol, UDP length	Source & Dest. Port	Data length & checksum	IR header	1024 bytes data

(c) Inter-frame packet format

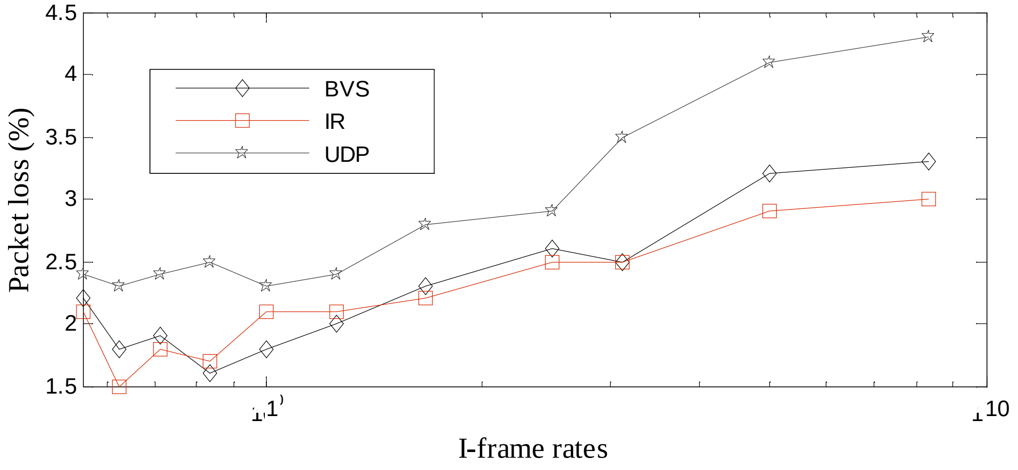
Figure 8.2: Packet format of the transport protocols

For each experiment, a 10-second MPEG4 video trace, Akiyo [32], is streamed 50 times. Since the default java UDP streamer considers error packets as lost packets, the java video streamer was simplified by replacing video data with random bytes in order to minimize the video reading time involvement in end-to-end protocol performances. All nodes were synchronized using a network time protocol (NTP); NetTime [141].

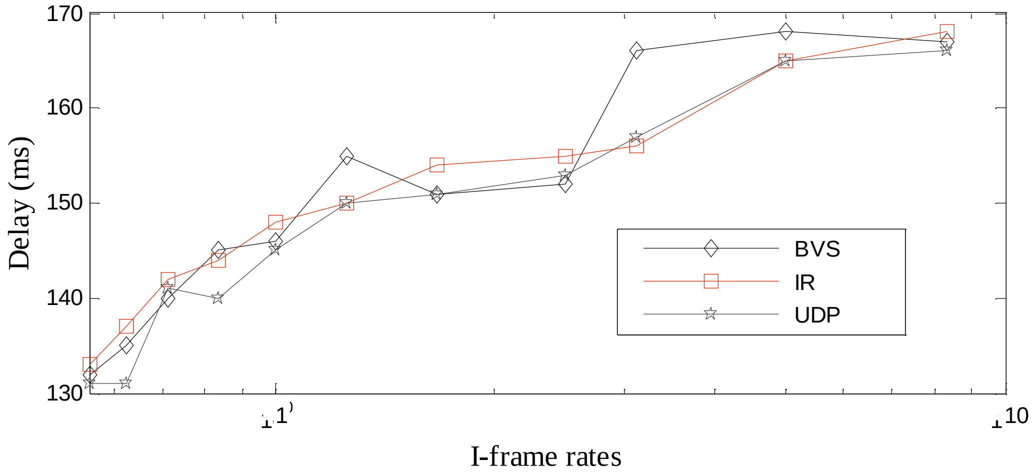
However, since the NTP accuracy varies between 10 and100 ms, it was difficult to compare packet delay from different computers. Therefore, only one video stream was evaluated. Experiments were repeated for different protocols and GOPs, and both packet loss and delay were measured.

8.3.2 Experiment results

The experiment goal is to further strengthen the simulation results in Section 5.2 and mathematical analysis in Chapter 7. Although the performances of the protocols are affected by the design of the software, the results present a sample of experimental approaches. Collected from 50 repetitions for each GOP, the experiment results are presented in Figure 8.3.



(a) Packet loss



(b) Packet delay

Figure 8.3: Protocol performance comparisons

Packet loss characteristics show that both NACK-based protocols are superior to UDP. The proposed IR protocol performs better than quick response, especially when I-frame rates are greater than 1.25. For I-frame rates smaller than 1.25, the three protocol performances vary.

Figure 8.3b shows the packet delay comparison among the three protocols. Since video encoding and decoding process are not included, end-to-end delays are fewer than 170ms. UDP dominates the lowest delay over almost all I-frame rates. Both BVS and IR protocol delays vary interchangeably. The delay differences are not significant due to the high bandwidth availability. However, the average delay of IR is slightly lower than BVS protocol, with values of 150.18 s and 150.64 s respectively.

8.4 Transport layer protocol performance in 802.11 networks

Since 802.11 networks are used widely in existing surveillance networks, the performance of the proposed protocol in 802.11 networks was also examined.

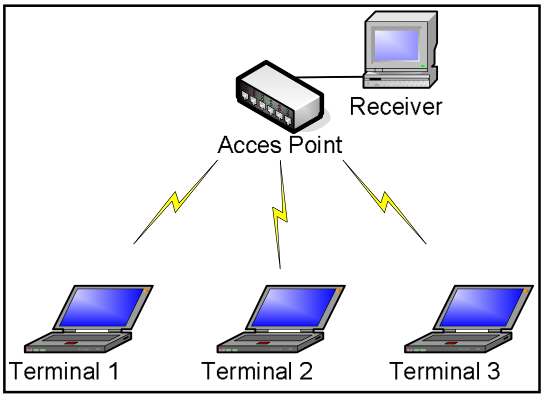
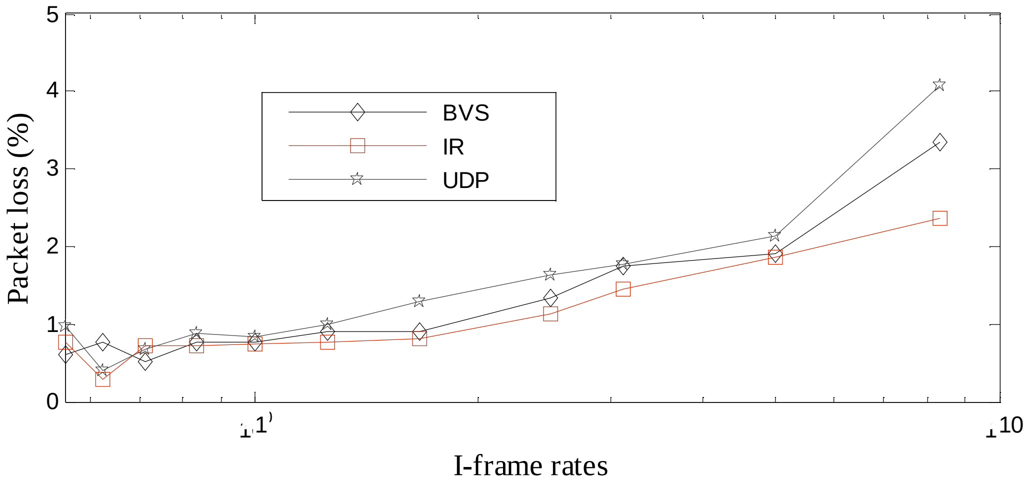


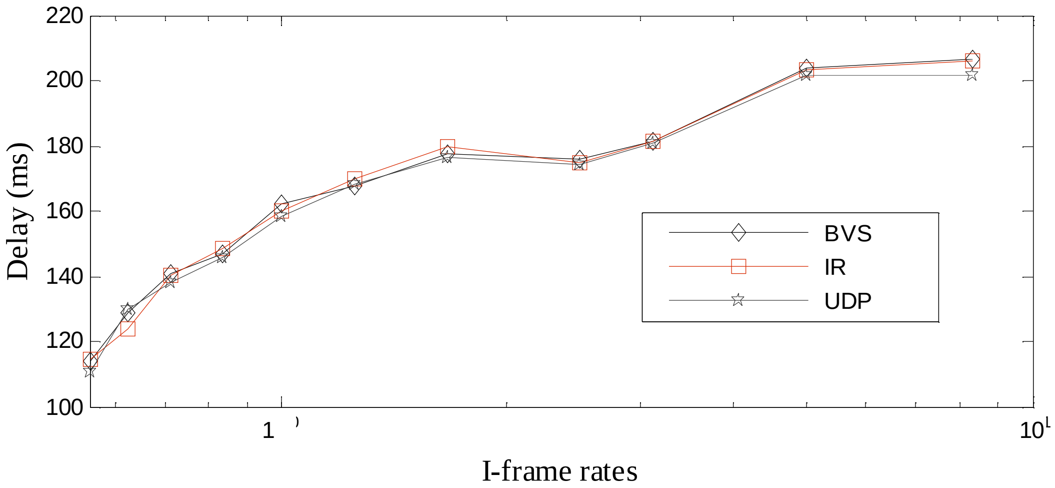
Figure 8.4. Diagram of the experimental 802.11 network

Experiments were conducted in a static environment where no other 802.11 network existed nearby. The assessment used the streamer design outlined in Section 8.3. The stream receiver was connected through the ethernet port of a TP-Link device, as illustrated in Figure 8.4. Three mobile computers sent video data wirelessly to the receiver. The experiment used 2 out of 3 mobile computers to generate CBR streams as background traffic, and one mobile computer to generate two video streams. The latter is evaluated.

The average packet loss and delay values for the 50 repetitions are presented in Figure 8.5. IR is able to reduce packet loss better than the existing protocols BVS and UDP. In terms of delay, IR and BVS delay differences are not significant. UDP remains superior to other protocols.



(a) Packet loss



(b) Packet delay

Figure 8.5: Packet loss and delay in an 802.11 network

8.5 Post-decoding error concealment techniques

8.5.1 Streamer design

The streamer employed well-known third-party software that is fast and easy to use; Ffmpeg, a cross-platform audio and video solution software [142] to process video. The Xuggle java wrapper software [143] bridges java programming language and Ffmpeg, enabling command execution in java code. Ffmpeg contains an encoder and decoder that are able to work on many video coding types. The packet is in a compressed form and rearranged to suit serialized transmission. Additional headers are added to differ whether video or audio packets, as error concealment to both streams, are different. Figure 8.6 demonstrates the header arrangement.

Packet Sequence	Video Sequence	Video Header	Data	Optional (Error Concealment)

Figure 8.6: Header format

Information regarding the error concealment method is inserted as an optional header. The header and data allocation are arranged either in bytes stream or as a serialized object. Byte arrangement results in less transmitted packet size, while serialized object provides more flexibility.

The streaming used UDP protocol and maximized the Maximum Transmission Unit (MTU) as no intermediate router involved. Once UDP packets are received, they will be extracted and analyzed. As the received packet is in transmission-suit packet form, it should be rearranged to its original video packet before being decoded. If dropped packets are found, the error concealment will be applied.

Table 8.1: The observed video in SS2

Parameter	Value
Sequence	Blue sky
Quality factor	31
Frame rate	25 fps
Video codec/sequence	MPEG4/IPP
Container	AVI
Bit rate	2631 kbps
Maximum transfer unit	64000 bytes
Number of frames	180
Resolution	1920x1080

Differing from research performed in [1, 136, and 137], the experiment focuses on evaluating the single traffic source in SS2 to deliver high bit rate video. The application is intended for video surveillance systems with a high bit rate video streaming application. The experiment uses blue sky, an HD video sequence taken from [152], with the codec details provided in Table 8.1. The video sequence was selected to suit a surveillance camera in either a fixed position or with fixed-period movement.

The system's background traffic in SS1 is video streaming using mpeg4 and mp3 codec in an AVI video container, transmitted using MoSES, UDP-based mobile streaming software designed by Gualdi [9] with a maximum bit rate of 200 kbps. The SS2 traffic is observed by repeating the experiment in several trials. The error concealment methods are applied if packet loss occurs. Lost packet and lost frame terms are used interchangeably as one frame fits into one packet. The number of frame loss, rate variation and PSNR video quality values are evaluated.

8.5.2 Experiment results

During the experiment, 12 trials of the streaming of 180 video frames were observed. The 180 frames are duplicated three times to be 540 frames in order to avoid the on-off transients and to mimic in-progress transmission. However, only the second 180 frame sequence was observed. Figure 8.7 draws from the 12 trials and shows three dropped packets in trial 3 frame-21, trial 6 frame-68 and trial 11 frame-115. This means that the packet loss rates in those trials are 1 of 180 packets (0.55%), while for the other 9 trials, the packet loss rates are 0%.

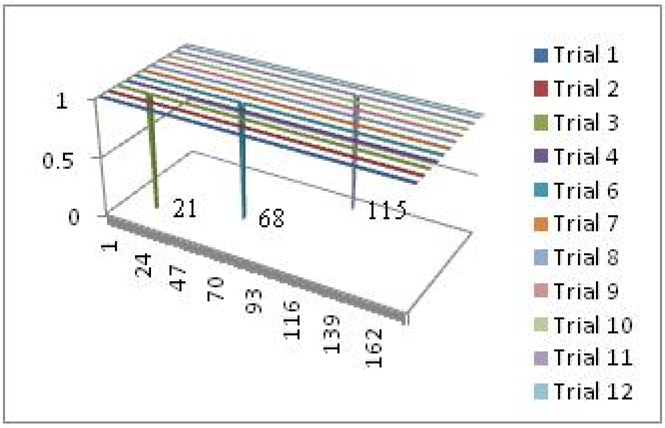


Figure 8.7: The packet loss pattern; '1' = accepted, '0' = lost.

This outcome confirms that the high MTU size does not always lead to higher packet loss; moreover, the examined link is point-to-point connection. It was also demonstrated by [136] that the smaller the number of concurrent connections, the smaller the number of dropped packets.

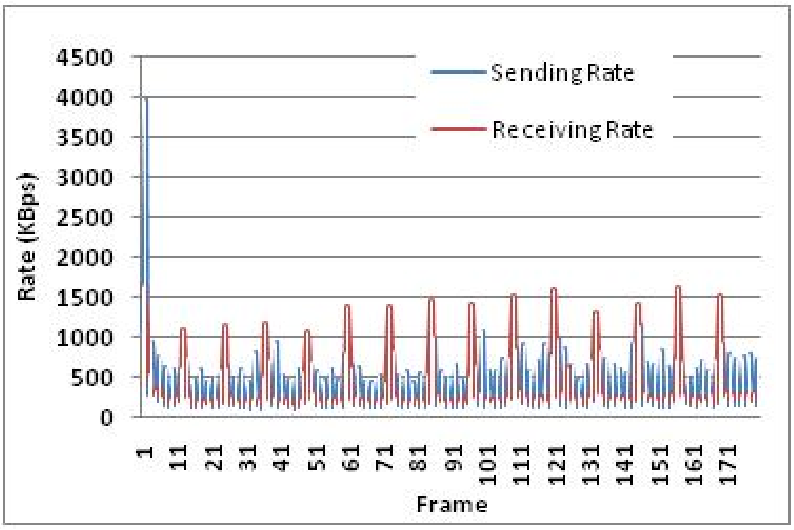


Figure 8.8: Rate variation

The rate variation in each transmitted/received frame was measured as the number of transmitted/received bytes divided by the sent or received time differences between two consecutive frames.

Figure 8.8 depicts the average rates for transmitted/received frames in all trials. I-frames generate more bytes, producing higher rates for both sending and receiving rates, while P-frames have smaller bytes, producing smaller rates. P-frames have a higher sending rate than the receiving rate due to the higher frame processing time for the receiver than the sender. The receiver should move the received packets from the UDP buffer to the application memory, as the UDP buffer size is limited.

Conversely, when processing I-frames, the receiving rate is higher than that of the sending rate. The processing time in the receiver is not sensitive to packet size; a higher frame size within the received packet leads to a higher receiving rate. This demonstrates that the I-frames utilize the WiMAX link better than P-frames.

The average sending rate is 470.35 kBps or 3762.8 kbps and the average receiving rate is 340.81 kBps or 2726.48 kbps. This means that the link is able to accommodate the transmitted 2631 kbps video. In relation to the lost packet, it is not caused necessarily by the network; the end node characteristics, such as processor, may also contribute [101]. The post-decoding lost frame error concealment methods are applied to dropped packets. The PSNR performance of each concealed packet is given in Table 8.2.

Table 8.2: PSNR Performance of post-decoding error concealment

Concealed frame	Non-Error	Error concealment methods					
		FC	Inline	Referenced	CR	Combine	Pattern
21	35.1205	26.18122	35.1205	33.0793038	26.18122	26.199558	29.74231
68	37.02699	23.80111	37.02699	33.8171658	24.27474	25.506293	27.71688
115	36.07375	24.99117	36.07375	33.4482348	25.22798	25.852925	28.72959

The inline frame copy method applied to fixed and fixed-motion camera replaces the lost packets using a previous co-located frame. Consequently, when no change is detected, the concealed frame quality is similar to an error-free condition. The CR method performs the same or better than FC method. While combined and patterned partial FC methods generate better PSNR performances than basic FC and CR, the referenced partial FC is superior to other methods; however, the method adds more payloads to UDP packets.

8.6 Summary

This chapter has presented an experimental assessment of the proposed methods, IR protocol and post-decoding error concealment techniques, to improve video surveillance applications over a WiMAX network.

The experiment has shown that even though WiMAX offers high bandwidth connectivity, packet loss still exists. The experimental work has proven the positive outcome of the proposed methods in reducing packet loss.

IR is able to reduce packet loss better than the existing BVS protocol. Although delay of both protocols is not significant, this is as a result of the high bandwidth availability. The experiment also showed that IR not only works well in a WiMAX network, but also in a 802.11 environment.

The experiment demonstrated that post-decoding lost frame concealment methods can be used as the last method of improving the PSNR of the received video. The methods increase PSNR up to 10 dB over frame copy method for non-static images. Although those frame copy-based methods improve only the PSNR of lost frame, their PSNR performance improvement are comparable to MV methods.

Chapter 9
Conclusion

9.1 Summary

In this thesis, multi-layer improvements on WiMAX based video surveillance have been investigated, including MAC, transport and application layers. To increase the surveillance performance, some methods have been proposed in each layer.

The evaluation of the existing and proposed methods is performed using simulation tools. The NS-2 simulator with an external NIST WiMAX module has been used to evaluate MAC and transport layer techniques, while JM15.0 H.264/AVC reference software was employed to evaluate the application layer technique. Mathematical analysis is also provided for transport layer protocols. To verify the simulation and analysis results, experimental works were conducted for transport and application layers.

MAC layer adjustment, which includes service class architecture, bandwidth request mechanism and scheduling algorithms, has been developed in Chapter 4. All SSs in a dedicated surveillance network generate video traffic. Normally, only the rtPS service class is utilized, while the UGS, ertPS, nrtPS and BE are not used. However, since the rtPS service uses the unicast bandwidth request, the polling service degrades the overall performance. Instead of using rtPS service class, this thesis suggests BE service class is employed to serve all SSs; the UGS, ertPS, nrtPS and rtPS are deactivated. The revised service architecture is referred as to as flat BE service. The proposed architecture successfully decreases packet delay up to 1.06s and reduces packet loss up to 38%.

In order to benefit from the uniform SS requirements that have similar packet characteristics, the bandwidth request mechanism and the scheduler algorithms are designed to prioritize important packets. This thesis considers that I-frame packets are prioritized, which means the I-frame packet from one SS is considered more important than the non-I-frame packet of the other SS. The proposed bandwidth request mechanism behaves differently to those packets. A contention bandwidth request is employed to serve prioritized packets and a piggybacking method is adopted for non-prioritized packets. Since the competitors in contention request are reduced (only for prioritized packets), the contention window is deducted to reduce request delay. Since the non-prioritized frame size varies, a piggybacked bandwidth request uses the average frame size value. This approach reduces the bandwidth request delay successfully, on average, 39.4 ms lower than contention request and 3.22 ms lower than piggybacking method.

The scheduler algorithm in BS determines which node is granted bandwidth. Since the I-frame packet is prioritized, the proposed schedulers directly allocate bandwidth when the request is for I-frame packets; otherwise, the request is postponed until there are no more requests for the same allocation. In order to reduce scheduling time, the proposed scheduler avoids sorting process. Therefore, RR and FIFO are modified as neither are sorting schedulers. The proposed scheduling algorithms are referred to as RR and FIFO-based non-sorting packet-aware schedulers. The proposed schedulers are able to maintain the quality of the received video as I-frame packets receive higher bandwidth priority. On average, the proposed RR and FIFO based schedulers improve PSNR performance about 0.94 dB and 1.18 dB over the frame based scheduler.

Transport layer improvement is developed in Chapter 5. An effective NACK scheduling is proposed to enhance retransmission protocol. Further, congestion delay, which prioritizes the retransmitted packet, is applied as a simple congestion control. The proposed protocol is called the IR protocol. Evaluations in both WiMAX and WiFi networks reveal the capability of the proposed protocol by reducing packet delay up 10 ms and packet loss up to 0.15% over the existing retransmission protocol.

In order to allocate sufficient bandwidth for the retransmitted packets, an early bandwidth request mechanism in MAC layer is introduced. The bandwidth increment is achieved by adding the retransmitted bytes from the NACK packet to the earliest bandwidth request. This method is able to improve the performance of IR and BVS protocols significantly. On average, the proposed method decreases IR delay at least 14.3% and BVS delay at least 12.6 %. The method also increases video performances of IR protocol 2.22 dB and BVS protocol 5.13 dB.

Mathematical analysis that combines application, transport and MAC layers is proposed in Chapter 7. The application layer stream is modelled in a two-state Markov model. Packet arrangement is adjusted according to the transport layer protocol. This model is then inserted in the WiMAX model, whereby packet loss and delay are derived.

Application layer improvement is developed by implementing the rateless code and post-decoding error concealment techniques detailed in Section 6. Rateless code is applied as an application layer redundancy with transmission time adjustment. The results show that the rateless code is able to improve the video performance, mainly when the channel is erroneous. On average, the rateless code increases TMC video performance about 1.34 dB. Post-decoding error concealment techniques are considered the last means by which to tackle packet loss. The proposed error concealment techniques are based on the frame copy method. The techniques can be combined with existing methods as the proposed methods are applied just before the image is rendered on-screen. The methods increase FC performance up to 6.64 dB.

Assessment of the selected proposed methods in an experimental network is described in Chapter 8. The experiment evaluations were applied to transport layer protocols and error concealment techniques. The experiments were limited to methods that do not involve MAC layer because this type of access to existing WiMAX devices is restricted. The assessments confirm the performance yielded from the simulation and mathematical analysis.

9.2 Future work

Plenty of opportunities remain to enhance the performance of WiMAX as a surveillance network. The following sections outline several suggestions for future work.

9.2.1 MAC layer

The proposed bandwidth request mechanism implements a contention request for I-frame. Contention window modification is chosen to minimize deviation from the standard. However, many works suggest alternative methods that differ greatly from the standard, especially if it is applied by the hardware manufacturer. Therefore, the opportunity to increase bandwidth request performance is still wide open.

When designing the schedulers, it is assumed that all nodes have similar video characteristics. However, video surveillance generated by SS may vary, as each surveillance point may have different requirements. For instance, a fixed camera may satisfy one surveillance area, but others may require a moving camera. Room surveillance may need lower positioning than street surveillance. Camera position may require rate adjustment. Considering these characteristics when designing the schedulers may increase WiMAX performance for video surveillance application.

9.2.2 Transport layer protocol

This thesis evaluates transport layer protocol performance in WiMAX network with a single BS. Surveillance network may consist of only one BS, but it can also be in the form of networks. SSs may move from one BS to another. The performance of the proposed protocol could be improved for wider networks that involve handover and routing processes.

9.2.3 Application layer techniques

Post-decoding error concealment techniques are implemented based on the frame copy method. This method was chosen because of its simplicity, and it was considered as a final solution before the video is rendered on-screen. Most error concealment techniques focus on MV. Implementing the proposed method, mainly techniques that require processing prior transmission, in MV might increase the performance of the error concealment technique.

Furthermore, as outlined in Section 2.6, application layer techniques are not limited to those developed in Chapter 7. Other methods in the application layer may be able to contribute to the performance enhancement of a WiMAX-based surveillance system.

References

[1] Scalabrino, N., De Pellegrini, F., Riggio, R., Maestrini, A., Costa, C.; Chlamtac, I., *Measuring the Quality of VoIP Traffic on a WiMAX Testbed,* in the proceeding of the 3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007.

[2] Vu H. L., Chan S., Andrew L., *Performance Analysis of Best-Effort Service in Saturated IEEE 802.16 Networks,* IEEE Transactions on Vehicular Technology, vol. 59, no. 1, pp. 460-472, Jan. 2010.

[3] Suherman S., Al-Akaidi M., "Adjusting WiMAX for a dedicated surveillance network", International Journal of Electrical and Computer Engineering, vol 3, no 4, pp. 492-503, August 2013.

[4] Suherman S., Al-Akaidi M., Hamzaoui R., *Inter-frame Retransmission for Video Surveillance over WiMAX*, in the proceeding of the 12th Middle Eastern Simulation and Modeling Conference, Amman, Jordan, pp 42-47, November 2011.

[5] Suherman S., Al-Akaidi M., *An efficient negative acknowledgement-based transport protocols in 802.11 media streaming*, International Journal of Ad Hoc and Ubiquitous Computing (accepted on 22nd April 2013).

[6] Suherman S., Al-Akaidi M., *Increasing Uplink Broadband Video Streaming Performance in WiMAX Networks*, International Journal of Internet Protocol, vol. 7 no. 3, pp. 176-185, 2012.

[7] Suherman S., Al-Akaidi M., *Post-decoding Concealment Techniques for Video Transmission*, in the proceeding of the Global Conference of Communication Science and Information Engineering, London, July 2011.

[8] Golembewski, R., Rossberg, M., Schaefer, G., *Towards Smart Infrastructures for Modern Surveillance Networks,* Technical report, TU-Ilmenau,2011.

[9] Gualdi G., Prati A., Cucchiara R., *Video Streaming for Mobile Video Surveillance,* IEEE Transaction on Multimedia, Vol. 10, No. 6, October 2008.

[10] Zhao, Y., *Design and implementation of video surveillance system based on CDMA cellular wireless networks,* in Proc. International Conference on Information and Automation, June 2008.

[11] Imai Y., Hori Y., & Masuda S., *Development and a brief evaluation of a web-based surveillance system for cellular phones and other mobile computing clients*, in Proc. IEEE Conference on Human System Interactions, pp. 526-531, 2008.

[12] IEEE Std. 802.16-2004, *Local and Metropolitan Area Networks*, *Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE, October 2004.

[13] IEEE Std. 802.16e-2005, *Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems*, IEEE, December 2005.

[14] IEEE Std 802.16m-2011(Amendment to IEEE Std 802.16-2009), *IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems Amendment 3: Advanced Air Interface,* IEEE, pp.1-1112, May 5, 2011.

[15] Stallings, W., *Data and Computer Communications 7th Edition*, Prentice Hall, Upper Saddle River, NJ, 2000.

[16] Denenberg, R., *Open systems interconnection*, Library hi tech, 3(1), 15-26, 1985.

[17] Park E. C., *Efficient Uplink Bandwidth Request with Delay Regulation for Real-Time Service in Mobile WiMAX Networks*, IEEE Transactions on Mobile Computing, vol.8, no.9, pp.1235-1249, Sept. 2009.

[18] Nair G., et al., *IEEE 802.16 Medium Access Control and Service Provisioning*, Intel Technology Journal, vol. 8, no. 3, pp. 213- 228, 2004

[19] Noordin K. A., Markarian G., *Providing QoS support through scheduling in WiMAX systems*, International Journal of the Physical Sciences, vol. 6(16), pp. 4070-4081, August 18, 2011.

[20] Kang S.H., Zakhor A., *Packet scheduling algorithm for wireless video streaming*, in Proc. International Packet Video Workshop, Pittsburgh, PY, April 2002.

[21] Wang Q-L., Liu G-Z., *A priority-based EDF scheduling algorithm for H.264 video transmission over WiMAX network*, in Proc. IEEE International Conference on Multimedia and Expo, pp.1-6, July 11-15, 2011.

[22] Chi-Hong J., Tzu-Chieh T., *Token bucket based CAC and packet scheduling for IEEE 802.16 broadband wireless access networks*, in Proc. 3rd IEEE Consumer Communications and Networking Conference, pp. 183-187, 2006.

[23] Chandra S., Sahoo A., *An Efficient Call Admission Control for IEEE 802.16 Networks*, in Proc. 15th IEEE Workshop on Local & Metropolitan Area Networks, pp. 188-193, 2007.

[24] Agis E. et al., *Global, Interoperable Broadband Wireless Networks: Extending WiMAX Technology to Mobility*, Intel Technology Journal, 8(3):173-187, 2004.

[25] Hu R. Q., et al., *On the Evolution of Handoff Management and Network Architecture in WiMAX*, in Proc. IEEE Mobile WiMAX Symposium, pages 144-149, Florida, USA, 25-29 March 2007.

[26] Lee D. H., Kyamakya K., Umondi, J. P., *Fast Handover Algorithm for IEEE 802.16e Broadband Wireless Access System*, in Proc. 1st International Symposium on Wireless Pervasive Computing, Phuket, Thailand, 16-18 January 2006

[27] Rouil R., Golmie N., *Adaptive Channel Scanning for IEEE 802.16e*,in Proc. IEEE Military Communications Conference (MILCOM), pages 1-6, Washington D.C, USA, 23-25 October 2006.

[28] Choi S., et al., *Fast handover scheme for Real-Time Downlink Services in IEEE 802.16e BWA System*, in Proc. Vehicular Technology Conference (VTC), vol. 3, pages 2028-2032, Stockholm, Sweden, Spring 2005.

[29] Kim, J., Kim, D. E., Park, S., Rhee, S. H., Han, K., Kang, H, *Use of vestigial power management bit within Wi-Fi frame structure of access point for coexistence of Wi-Fi and WiMAX systems in shared bands*, in Proc. IEEE International Conference on Ubiquitous and Future Networks, pp. 220-224, June 2009.

[30] Zheng, R., Zhang, X., Li, X., Hai, Y., Yang, D., *Coexistence Study in the 2500-2690 MHz Band between WiMAX and WCDMA Systems*, in Proc. IEEE 68th Vehicular Technology Conference, pp. 1-6, Sept. 2008.

[31] Sartori, L., Elayoubi, S. E., Fourestié, B., Nouir, Z., *On the WiMAX and HSDPA coexistence*, in Proc. IEEE International Conference on Communications, pp. 5636-5641, June 2007.

[32] Tani, A., & Fantacci, R., *A low-complexity cyclostationary-based spectrum sensing for UWB and WiMAX coexistence with noise uncertainty*, IEEE Transactions on Vehicular Technology, Vol. 59, No. 6, pp. 2940-2950, 2010.

[33] Bo, F., Fang-jing, H., Ai-min, Z., Fang-jian, H., *Spectrum sensing in MB-OFDM based cognitive UWB systems coexistence with WiMAX*, in Proc. 11th IEEE Singapore International Conference on Communication Systems, pp. 203-207, November 2008.

[34] 3GPP TSG RAN, TR 25.913 v7.3.0, *Requirements for Evolved UTRA (E-UTRA)and Evolved UTRAN (E-UTRAN)*, March 2006.http://www.3gpp.org

[35] Postel J., *User Datagram Protocol*, Request for Comments RFC 768, ISI, August 1980.

[36] Larzon, L. K., Degermark, M., Pink, S., *UDP-lite for real time multimedia applications*, in Proc. IEEE International Conference of Communications, June 1999.

[37] Raisinghani, V. T., Iyer, S., *Cross-Layer Design Optimizations in Wireless Protocol Stacks*, Computer Communication (Elsevier) Journal, vol. 27, pp. 720–24, 2004.

[38] Noordin, K. A., Markarian, G., *Cross-Layer Optimization Architecture for WiMAX Systems*, in Proc. IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1-4, 2007.

[39] Vandalore, B., Feng, W. C., Jain, R., Fahmy, S., *A survey of application layer techniques for adaptive streaming of multimedia*, Real-Time Imaging, Vol. 7, No. 3, pp. 221-235, 2001.

[40] National Institute of Standards and Technology, *The Network Simulator ns-2 add-on*, IEEE 802.16 model (MAC+PHY), June 2007.

[41] Chen, Q., Schmidt-Eisenlohr, F., Jiang, D., Torrent-Moreno, M., Delgrossi, L., Hartenstein, H., *Overhaul of IEEE 802.11 modeling and simulation in ns-2*, in Proc. 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, pp. 159-168, October 2007.

[42] Chen J., Wang C. C., Tsai F. C. D., Chang C. W., Liu S. S., Guo J., Hung, C. H., *The design and implementation of WiMAX module for ns-2 simulator*, in Proc. workshop on ns-2: the IP network simulator, pp. 5, October 2006.

[43] Abhayawardhana, V. S.et al., *Comparison of empirical propagation path loss models for fixed wireless access systems*, in Proc. IEEE Vehicular Technology Conference, Vol. 1, pp. 73-77, May 2005.

[44] Xia, H., Bertoni, H. L., Maciel, L. R., Lindsay-Stewart, A., Rowe, R., *Radio propagation characteristics for line-of-sight microcellular and personal communications*, IEEE Transactions on Antennas and Propagation, Vol. 41, no. 10, pp. 1439-1447, 1993.

[45] Hari, K. V. S., Baum, D. S., Rustako, A. J., Roman, R. S., Trinkwon, D., *Channel models for fixed wireless applications*, IEEE 802.16 Broadband Wireless Access Working Group, 2003.

[46] COST Action 231, *Digital mobile radio towards future generation systems*, Final Report - Directorate-General Telecommunication, Information society, Information Market, and Exploitation of Research, 1999.

[47] Athanasiadou, G. E., Nix, A. R., & McGeehan, J. P., *A microcellular ray-tracing propagation model and evaluation of its narrow-band and wide-band predictions*, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 3, pp. 322-335, 2000.

[48] Haßlinger, G., Hohlfeld, O, *The Gilbert-Elliott model for packet loss in real time services on the Internet*, in Proc. Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), pp. 1-15, March 2008.

[49] Klaue, J., Rathke B., Wolisz A., *Evalvid - A Framework for Video Transmission and Quality Evaluation*, in Proc. the 13th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation, 2003.

[50] Ke C-H., Shieh C-K., Hwang W-S., Ziviani A, *An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission*, Journal of Information Science and Engineering, 2008.

[51] ITU-T, *Methods for subjective determination of transmission quality*, Recommendation P.800, 1996.

[52] Test media. Retrieved from: http://media.xiph.org, accessed in May 2011.

[53] Ahmad I., Habibi D., *A WiMAX solution for real-time video surveillance in public transport*, International Journal of Computer Networks & Communications Vol.3, No.2, March 2011.

[54] Chang R-Y., Chiao H-T., Lee C-L., Chen Y-C., *Applications for IP Video Surveillance over the ITRI MTWAL*, Journal of networks, Vol. 5, No. 8, August 2010.

[55] Aguado, M., Jacob, E., Matias, J., Conde, C., Berbineau, M., *Deploying CCTV As an Ethernet Service Over the WiMAX Mobile Network in the Public Transport Scenario*, in Proc. IEEE International Conference on Communications, pp. 1-5, June 2009.

[56] Luo Y-M., Wei B., Jin Z-G., *SIP-based WIMAX Wireless Video Surveillance System*, in Proc. International Conference on Computer Science and Service System, pp. 1170-1173, August 2012.

[57] Lamy-Bergot C., Renan, E., Gadat B., Lavaux D., *Data supervision for adaptively transcoded video surveillance over wireless links*, in Proc. International Conference on Intelligent Transport Systems Telecommunications, pp.415-419, 20-22 Oct. 2009.

[58] Kim S., Yeom I., *Performance analysis of best effort traffic in IEEE 802.16 networks*, Tech. Rep. CS-TR-2008-285, Dept. Comput. Sci., Korea Advanced Inst. Sci. Technol., 2006.

[59] Iyengar, R., Sikdar, B., *A Queuing Model for Polled Service in WiMAX/IEEE 802.16 Networks*, IEEE Transactions on Communications, Vol. 60, No. 7, pp. 1777-1781, 2012.

[60] Zhu P., Zhu G-X., Lin H-Z., Shi H-B., *Adaptive scheduling strategy for WiMAX real-time communication*, in Proc. International Symposium on Intelligent Signal Processing and Communication Systems, pp.718,721, November 2007

[61] Fallah, Y. P., Agharebparast, F., Minhas, M. R., Alnuweiri, H. M., Leung, V. C., *Analytical modeling of contention-based bandwidth request mechanism in IEEE 802.16 wireless networks,* IEEE Transactions on Vehicular Technology, Vol. 57, No. 5, pp.3094-3107, 2008.

[62] Ni Q., Hu L., Vinel A., Xiao Y., Hadjinicolaou M., *Performance Analysis of Contention Based Bandwidth Request Mechanisms in WiMAX Networks,* IEEE Systems Journal, vol.4, no.4, pp.477-486, December 2010.

[63] Vinel A., Zhang Y., Lott M., Tiurlikov A., *Performance analysis of the random access in IEEE 802.16*, in Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun., Berlin, Germany, pp. 1596–1600, Sep. 2005.

[64] Iyengar R., Iyer P., Sikdar B., *Analysis of 802.16 based last mile wireless networks,* in Proc. IEEE Global Telecommunications Conference, vol.5, no., p.5 and p.3127, Dec. 2005.

[65] Chuck D., Kuan-Yu C., Chang J. M., *A Comprehensive Analysis of Bandwidth Request Mechanisms in IEEE 802.16 Networks*, IEEE Transactions on Vehicular Technology, vol.59, no.4, pp.2046-2056, May 2010.

[66] Chen L-W., Tseng Y-C., *Design and analysis of contention-based request schemes for best-effort traffics in IEEE 802.16 networks,* IEEE Communications Letters, vol.12, no.8, pp.602-604, Augustus 2008.

[67] He J-H., Yang K., Guild K., Chen H-H., *On bandwidth request mechanism with piggyback in fixed IEEE 802.16 networks*, IEEE Transactions on Wireless Communications, vol.7, no.12, pp.5238-5243, December 2008.

[68] Kwak B-J., Song N-O., Kwon D. S., *Enhancement of IEEE 802.16 WirelessMAN Ranging Performance with EIED Backoff Algorithm*, in Proc. IEEE 66th Vehicular Technology Conference (VTC-2007), pp. 1902 – 1906, October 2007.

[69] Song N-O., Kwak B-J., Song J., Miller L. E., *Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm*, in Proc. VTC 2003-Spring, Vol. 4, pp. 2775–2778, April 2003.

[70] Rajesh A., Nakkeeran R., *MSCR based bandwidth request mechanism for heterogeneous WiMAX networks*, Procedia Engineering, Vol. 30, pp. 482-490, 2012.

[71] Thapa, A., Shin, S., *Utility Based Backoff (UBB) Algorithm for Initial Ranging Procedure in WiBro*, in Proc. IEEE 69th Vehicular Technology Conference, pp. 1-5, April 2009.

[72] Chou S-F., Liu J-H., Chao H-L., Guo T-C., Liu C-L., Tsai F-J., *Performance enhancement of contention-based bandwidth request mechanism in IEEE 802.16 WiMAX networks*, in Proc. IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications, pp.1287-1292, 26-30 September 2010.

[73] Mukul R., Singh P., Jayaram D., Das D., Sreenivasulu N., Vinay K., Ramamoorthy A., *An adaptive bandwidth request mechanism for QoS enhancement in WiMAX real time communication*, in Proc. IEEE International Conference on Wireless and Optical Communications Networks, pp. 5-10, April 2006.

[74] Liu C. Y., Chen, Y. C., *An Adaptive Bandwidth Request Scheme for QoS Support in WiMAX Polling Services*, in Proc. IEEE 28th International Conference on Distributed Computing Systems, pp. 60-65, June 2008.

[75] Nie C., Venkatachalam M., Yang X., *Adaptive polling service for next-generation IEEE 802.16 WiMAX networks*, in Proc. IEEE Global Telecommunications Conference, pp. 4754-4758, November 2007.

[76] Lee N., Lee S., Kim N., *A Fast Bandwidth Request Scheme in IEEE 802.16e OFDMA/TDD Systems*, in Proc. IEEE Third International Conference on Mobile

Ubiquitous Computing, Systems, Services and Technologies, pp.173-178, p.11-16, October 2009.

[77] Pries R., Staehle D., Marsico D., *Performance evaluation of piggyback requests in IEEE 802.16*, in Proc. IEEE VTC—Fall, Baltimore, pp. 1892–1896, Sep. 30–Oct. 3, 2007.

[78] Dhrona, P., Ali N. A., Hassanein H., *A performance study of scheduling algorithms in Point-to-Multipoint WiMAX networks*, in Proc. Local Computer Networks Conference, (Montreal, Que., Oct.) , pp. 843-850, 2008.

[79] Wongthavarawat K., Ganz A., *Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems,* International Journal of CommunicationSystems, Vol. 16, No. 1, pp. 81-96, February 2003.

[80] Vinay K., Sreenivasulu N., Jayaram D., Das D., *Performance evaluation of end-to-end delay by hybrid scheduling algorithm for QoS in IEEE 802.16 network*, in Proc. International Conference on Wireless and Optical Communication Networks, April 2006.

[81] Hwang J. Y., Han Y., *An adaptive traffic allocation scheduling for mobile WiMAX*, in Proc. IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1-5, Sept. 2007

[82] Ku J. M., Kim S. K., Kim S. H., Shin S., Kim J. H., Kang C. G., *Adaptive delay threshold-based priority queuing scheme for packet scheduling in mobile broadband wireless access system*, in Proc. Wireless Communications and Networking Conference, Vol. 2, pp. 1142-1147, April 2006.

[83] Oktay M., Mantar H. A., *A distance-aware scheduler for real-time applications in 802.16 WiMAX networks,* in Proc. First International Conference on Networked Digital Technologies, pp. 435-440, July 2009.

[84] Noordin, K. A., Markarian G., *Cross-Layer Optimization Architecture for WiMAX Systems*, in Proc. 18th International Symposium on Personal, Indoor and Mobile Radio Communications, Athens, pp. 1-4, September 2007.

[85] Farhadi R., Vakili V. T., Moghaddam S.S., *A Novel Cross-Layer Scheduling Algorithm for OFDMA-Based WiMAX Networks*, Int. Journal Communications, Network and System Sciences, Vol. 4, pp. 98-103,February 2011.

[86] Lin W., Zhang L-Y., Wang R., *A Novel and Effective QoS Guarantee Scheme in WiMAX System*, in Proc. International Conference on Measuring Technology and Mechatronics Automation, 2010.

[87] Song J-F., Li J-D., Li C-L., *A Cross-layer WiMAX Scheduling Algorithm based on Genetic Algorithm*, in Proc. 7th Annual Communications Networks and Services Research Conference, 2007.

[88] Oh S-M., Cho S-Y., Kim J-H., Kwun J-Y., *VoIP Scheduling Algorithm for AMR Speech Codec in IEEE 802.16e/m System*, IEEE Communication letters, Vol. 12, No. 5, May 2008.

[89] Zhang H., Chen H., Fu X., Dai J., *A scheduling algorithm based on channel state for VoIP services in IEEE Std 802.16 e system*, in Proc. IEEE 10th International Conference on Advanced Communication Technology, Vol. 3, pp. 1521-1525, January 2008.

[90] Brahmia M. A., Abouaissa A., Lorenz P., *Adaptive scheduling mechanism for IPTV over WiMAX IEEE 802.16 j networks*, International Journal of Communication Systems, Vol. 1, 2012.

[91] Huang S-M, Huang C-W, Wu P-H., Hwang J-N., Gau V., Chen Y-C., *Resource Efficient Opportunistic Multicast Scheduling for IPTV over Mobile WiMAX*, in Proc. IEEE Vehicular Technology Conference, pp. 1-5, May 2010.

[92] Sabri A, El-Badawy H., Shehata K., Ali A., *Video Scheduling in the Uplink over WiMAX Networks by the Novel Concept Adaptive Minimum Service Interval*, in Proc. International Conference on Computer Engineering and Applications, Vol.2, pp. 3-7, March 2010.

[93] Wu Y., Le Y., Zhang D., *An Internal Scheduling Framework for Management of Best Effort Connections per SS in WiMAX System*, in Proc. IEEE Global Telecommunications workshop, pp. 1-6., Nov. 30 -Dec. 4 2010.

[94] Sentinelli A., Marfia G., Gerla M., Kleinrock L., Tewari S., *Will IPTV ride the peer-to-peer stream?[Peer-to-Peer Multimedia Streaming]*, IEEE Communications Magazine, Vol.45, No.6, pp.86-92, June 2007.

[95] Atici C., Sunay M. O., *Improving the performance of wireless H.264 video broadcasting through a cross-layer design*, in Proc. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, pp.1-6, 13-15 May 2009.

[96] Zheng H., Boyce J., *An improved UDP protocol for video transmission over Internet-to-wireless networks*, IEEE Transactions on Multimedia, Vol.3, No.3, pp.356-365, September 2001.

[97] Bova T., Krivoruchka T., *Reliable UDP protocol*, IETF draft, Network Working Group, February 1999.

[98] Kohler E., Handley M., Floyd S., Padhye J., *Datagram congestion control protocol (DCCP)*, Request for Comment (RFC) 4340, IETF, March 2006.

[99] Wirz R., Marin R., Ferre M., Barrio J., Claver J. M., *Bidirectional transport protocol for teleoperated robots*, IEEE Transactions on Industrial Electronics, Vol.56, No. 9, pp. 3772-3781, September 2009.

[100] Gu Y. H., Grossman R. L., *UDT: UDP-based data transfer for high-speed wide area networks*, Elsevier Computer Networks, Vol. 51, No. 7, pp. 1777-1799, May 2007.

[101] He E., Leigh J., Yu O., Defanti T. A., *Reliable Blast UDP: predictable high performance bulk data transfer*, in Proc. IEEE International Conference on Cluster Computing, p.317, September 2002.

[102] Ali I., Al-Majeed S., Fleury M., Ghanbari M., *Semi-reliable transport protocol for IPTV over mobile WiMAX*, In Proc. IEEE International Conference on Computer as a Tool, pp. 1-4, April 2011.

[103] Le T., Kuthethoor G., Hansupichon C., Sesha P., Strohm J., Hadynski G., Kiwior D., Parker D., *Reliable User Datagram Protocol for airborne network*, in Proc. IEEE conference on military communications, pp. 1-6, October 2009.

[104] Lam P. P. K., Liew S. C., *UDP-Liter: An improved UDP protocol for real-time multimedia applications over wireless links*, in Proc. IEEE international symposium on Wireless Communication Systems, pp. 314-318, September 2004.

[105] Fox R., *TCP big window and NAK operations*, IETF RFC1106, June 1989.

[106] Tezcan N., Cayirci E., Caglayan M. U., *End-to-end reliable event transfer in wireless sensor networks*, in Proc. IEEE International Symposium on personal, indoor and mobile radio communications, Vol.2, pp. 989- 994, September 2004.

[107] Pajouhesh M., Bidgoli A. M., Yektaie M. H., *A New Reliable Data Delivery Protocol in Wireless Sensor Network*, Journal Basic Applied Science Research, Vol. 2, No. 1, pp.132-137, 2012.

[108] Adamson B., Bormann C., Handley M., Macker J., *NACK-Oriented Reliable Multicast (NORM) Transport Protocol*, Network Working Group, Request for Comments 5740, November 2009.

[109] Meddour D-E., Abdallah A., Ahmed T., Boutaba R., *A cross layer architecture for multicast and unicast video transmission in mobile broadband networks*, Journal of Network and Computer Applications (Elsevier), 2011.

[110] Martini, M. G., Hewage, C. T., Cross-layer frame synchronization for H. 264 video over WiMAX, in Proc. 2010 IEEE Eleventh International Workshop on Signal Processing Advances in Wireless Communications, pp. 1-5, June 2010.

[111] Al-Jobouri L., Fleury M., Ghanbari M., *Cross-layer scheme for WiMAX video streaming*, in Proc. Computer Science and Electronic Engineering Conference, pp. 86-91, July 2011.

[112] Yeh J. H., Chen J. C., Agrawal P., *Fast intra-network and cross-layer handover (FINCH) for WiMAX and mobile internet*, IEEE Transactions on Mobile Computing, Vol. 8, No. 4, pp. 558-574, 2009.

[113] Castro M. V., Fernandez D. P., *VoIP cross-layer load control for hybrid satellite-WiMAX networks*, IEEE Wireless Communications, Vol. 15, No. 3, pp. 32-39, 2008.

[114] Jo J., Cho J., *A cross-layer vertical handover between mobile WiMAX and 3G networks*, in Proc. Wireless Communications and Mobile Computing Conference, pp. 644-649, August 2008.

[115] Ye J., Wang J.-X., Huang J.-W, *A cross-layer TCP for providing fairness in wireless mesh networks*, International Journal of Communication Systems, Vol. 24, No. 12, pp. 1611–1626, December 2011.

[116] Zhai H., Chen X., Fang Y., *Improving Transport Layer Performance in Multi hop Ad Hoc Networks by Exploiting MAC Layer Information*, IEEE Transaction on Wireless Communications", Vol.6, No.5, pp.1692-1701, May 2007.

[117] Chien J-T., Li G-L.,Chen M-J., *Effective Error Concealment Algorithm of Whole Frame Loss for H.264 Video Coding Standard by Recursive Motion Vector Refinement*, IEEE Transactions on Consumer Electronics, Vol. 56, No. 3, August 2010.

[118] Chen C., Liu Y., Yang Z., Bu J-J., Deng X-Y., *Multi-frame Error Concealment for H.264/AVC Frames with Complexity Adaptation*, IEEE Transactions on Consumer Electronics, Vol. 54, No. 3, August 2008.

[119] Pyun J-Y., *Error concealment aware streaming video system over packet-based mobile networks*, IEEE Transactions on Consumer Electronics, Vol. 54, No. 4, pp.1705-1713, November 2008.

[120] Suissa A., Mellor J., Lohier F. R., Garda P., *A Novel Video Packet Loss Concealment Algorithm & Real Time Implementation*, in Proc. Conference on Design & Architectures for Signal & Image Processing (DASIP), 24-26 November 2008.

[121] Kim J-H., Han J-W., Cheon S-H., Baek S-J, Ko S-J., *A hybrid method for whole frame error concealment*, in Proc. IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pp. 406-409, 18-19 November 2009.

[122] Amiri, S.M., Bajic, I.V.,*A Novel Non causal Whole-Frame Concealment Algorithm for Video Streaming*, in Proc. IEEE International Symposium on Multimedia, pp.154-159, 15-17 December 2008.

[123] Tai S-C., Hong C-S., Fu C-A., *An Object-based Full Frame Concealment Strategy for H.264/AVC Using True Motion Estimation*, in Proc. The Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT), pp. 214-219, 14-17 Nov. 2010.

[124] Yan B., Gharavi, H., *Efficient error concealment for the whole-frame loss based on H.264/AVC*, in Proc. 15th IEEE International Conference on Image Processing, pp. 3064-3067, 12-15 Oct. 2008.

[125] Bin L., Hu J-L, Du M-H., *Error concealment for whole frame losses based on motion vectors estimation*, in Proc. The 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Vol. 6, pp. 358- 362, 20-22 Aug. 2010.

[126] Kim H-S., Ryu E-S., Jayant, N., *Channel-adaptive video transmission using H.264 SVC over mobile WiMAX network*, Digest of Technical Papers International Conference on Consumer Electronics (ICCE), pp. 441-442, 9-13 Jan. 2010.

[127] Maravedis, *WiMAX Statistics (Q3 2007) and Prospects*, IEC newsletter, February 2008.

[128] White Paper: *Fixed WiMAX Applications Focus: IP Video Surveillance Networks*, Fujitsu Microelectronics America, Inc. January 2007.

[129] White Paper: *Wireless IP Surveillance,* EWWP-010-1106, EION Inc., 2007

[130] Chughtai H. M. O., Malik S. A., Yousaf M., *Performance evaluation of transport layer protocols for video traffic over WiMAX*, in Proc. 3th IEEE Multitopic Conference, Dec 14-15, 2009.

[131] Luby M., Gasiba T., Stockammer T., Watson M., *Reliable download delivery in cellular broadcast networks*, IEEE Transactions on Broadcasting, Vol. 53, No. 1, pp. 235 – 246, 2007.

[132] Al-Jobouri L., Fleury M., Ghanbari M., *Adaptive rateless coding for IPTV over a mobile WiMAX channel,* in Proc. 8th Annual IEEE Consumer Communications and Networking Conference, 2011.

[133] H.264/AVC JM Reference Software.Retrieved from: http://iphome.hhi.de/suehring/tml/.

[134] Badia L., Baldo N., Levorato M., Zorzi M., *A Markov framework for error control techniques based on selective retransmission in video transmission over wireless channels*, IEEE Journal on Selected Areas in Communications, Vol. 28, No. 3, pp. 488-500, April 2010.

[135] Zhao J., Li B., Kok C., Ahmad I., *MPEG-4 video transmission over wireless networks: A link level performance study*, Journal of Wireless Networks, Vol. 10, No. 2, pp.133–146, March 2004.

[136] Ruiz G., Pubill D., Bader F., Ortega J.A., *Video Streaming in Uplink mode Using WiMAX System -- Experimental Results*, in Proc. Personal, Indoor and Mobile Radio Communication, 2009.

[137] Pentikousis K., Pinola, J., Piri, E., Fitzek, F., *An experimental investigation of VoIP and video streaming over fixed WiMAX*, in Proc. Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008.

[138] Issa, O., Wei L., Hong L., *Performance Evaluation of TV Over Broadband Wireless Access Networks*, IEEE Transactions on Broadcasting, Vol. 56, No. 2, pp. 201-210, June 2010.

[139] Halepovic E., Ghaderi M., Williamson C., *Multimedia application performance on a WiMAX network*, in Proc. Annual Multimedia Computing and Networking Symposium, San Jose, CA, January 2009.

[140] Aperto networks, Retrieved from http://www.apertonet.com.

[141] Puschner P., *Real-time performance of sorting algorithm*, Real-Time Systems, Vol. 16, No. 1, pp. 63-79, January 1999.

[142] Belghith A., Nuaymi L., *Design and implementation of a QoS-included WiMAX module for NS-2 simulator*, in Proc. of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, Brussels, Belgium, March 2008.

[143] Borin, J. F., Da Fonseca, N. L. S., *WiMAX module for the ns-2 simulator*, Technical Report, IC-11-27, Relatório Técnico, December 2011.

[144] Andreadis A., Rizzuto A., Zambon R., *A new NS-2 tool to investigate QoS management over mobile WiMAX*, in Proc. of the 4th International ICST Conference on Simulation Tools and Techniques, pp. 240-248, Brussel, Belgium, 2011.

Appendix 1

The NS-2 implementation

1. The modified files

In order to implement the proposed methods, changes should be performed in some NS-2 files, mainly the files that consist of C++ programs. The following is the list of files involved in the proposed method implementation.

- udp.cc and udp.h; both files consist of C++ scripts for UDP, both sender and receiver routines. Those files are modified and saved as IR.cc and IR.h.
- packet.h; this file accommodates codes for packet header.
- agent.cc and agent.h; the files register all agent used in NS-2.
- ns-default.tcl; it initiates default values for constants and variables.
- makefile.in; it contains lists of instructions and files for C++ compilation.
- mac.cc and mac.h; mac files represent medium access layer functionalities in NS-2.
- mac802_16BS.cc and mac802_16BS.h; these files consist of codes for NIST WiMAX BS module. The next listed files are parts of NIST WiMAX modules.
- mac802_16SS.cc and mac802_16SS.h; these files consist of codes for NIST WiMAX SS module.
- contentionrequest.cc and contentionrequest.h deal with contention request mechanism.
- wimaxscheduler.cc and wimaxscheduler.h; these files deal with BS and SS scheduling.
- bsscheduler.cc and bsscheduler.h; these files deal with BS scheduling.
- ssscheduler.cc and ssscheduler.h; these files deal with SSs scheduling.
- connection.cc and connection.h; these files contain C++ code for connections established between BS and SSs.

- Peernode.cc and peernode.h; these files contains C++ code for methods and variables used by SSs.

2. Transport layer protocol implementation

2.1. Inter-frame retransmission.

The proposed protocol script is realized by modifying the existing UDP C++ files in application directory /ns-2.31/apps. The new C++ files are IR.cc and IR.h. The resource ID in the opening scripts should be changed to protocol file name:

```
static const char rcsid[] = "@(#) $Header: /cvsroot/nsnam/ns-2/apps/IR.cc.
```

Before applying methods of the proposed protocol in IR.cc and IR.h, the proposed protocol should be introduced to ns simulator as a new transport layer agent. The agent is linked to ns program by adding the following scripts into packet.h file in directory /ns-2.31/common.

```
enum packet_t {
        PT_TCP,
        PT_UDP,
        PT_IR, //added
```

and

```
class p_info {
public:
        p_info() {
                name_[PT_TCP] = "tcp";
                name_[PT_UDP] = "udp";
                name_[PT_IR] = "IR"; //added
                name_[PT_CBR] = "cbr";
                name_[PT_AUDIO] = "audio";
```

In order to enable protocol execution, those IR.cc and IR.h should be registered in makefile.in for compilation proposed. The file names are added in script:

```
OBJ_CC = \
        tools/random.o
```

```
tools/rng.o
tools/ranvar.o
common/misc.o
common/timer-handler.o
apps/IR.o //added
```

New parameters are introduced in the proposed protocol. Those parameters can be categorized as common information or protocol headers depending upon the requirement. Common information regarding the application layer data such as frame_type, frame_size and packet_nb may be added to common packet header. The frame_type is to indicate what video frame the packet is part of. The frame_size shows the byte number of the video frame and packet_nb is the number of packets for transporting the frame. Those parameters are inserted in struct hdr_cmn in file packet.h which is located in directory /ns-2.31/common. The following is part of the modified code:

```
struct hdr_cmn {
        enum dir_t { DOWN= -1, NONE= 0, UP= 1 };
        packet_t ptype_;
        int     uid_;
        int     error_;
        int errbitcnt_;
        int frame_type;//added
        double frame_size;  // added
        double frame_nb; //added
```

Protocol header which contains some parameters is defined in file IR.h. Since the protocol introduces two packets: IR packet and NACK packet, there are two types of protocol headers: IR header and NACK header. The following scripts show the header implementation in IR.h file.

```
struct  hdr_IR  {
        u_int32_t srcid_;
        int seqno_;
        double sendtime_;
```

```
        double rectime_;
        int lastpacket_;
        int retryCount_;
        u_int32_t&srcid() { return (srcid_); }
        int& retryCount() { return (retryCount_); }
        static int offset_;
        inline static int& offset() { return offset_; }
        inline static hdr_IR* access(const Packet* p) { return (hdr_IR*) p->access(offset_);}
};
struct  hdr_NACK {
        u_int32_t srcid_;
        int seqno_;
        double sendtime_;
        double rectime_;
        int lastpacket_;
        int retryCount_;
        int losslist[20];
        u_int32_t&srcid() { return (srcid_); }
        int& retryCount() { return (retryCount_); }
        static int offset_;
        inline static int& offset() { return offset_; }
        inline static hdr_NACK* access(const Packet* p) {
        return (hdr_NACK*) p->access(offset_);}
};
```

The protocol headers are designed to carry information from both the sender and the receiver. In the proposed IR protocol, the parameter seqno_ determines packet sequence, sendtime_ and rectime_ are for delay calculation, lastpacket_ defines either packet is the last packet within a frame, retryCount flags either the retransmitted packet or not and losslist records indices of the lost packets.

IR class is declared as a new agent in IR.h.All constant and methods in IR.cc are defined in IR.h. Since IR is a transport layer agent, it extends class Agent.

```
class IRAgent;
class IRAgent : public Agent {
public:
	friend class IRAgent(); IRAgent(packet_t);
	virtual void sendmsg(int nbytes, const char *flags = 0)
	{
		sendmsg(nbytes, NULL, flags);
	}
	virtual void sendmsg(int nbytes, AppData* data, const char *flags = 0);
	virtual void recv(Packet* pkt, Handler*);
	virtual int command(int argc, const char*const* argv);
	virtual void addloss(int start, int last);
	virtual void resend();
	//additional public constants and variables
protected:
	//additional protected constants and variables
};
```

Default values for constants and variables should be determined. These values are declared in file ns-packet.tcl in directory ns-2.31/tcl/lib. The following is a sample of default value declaration.

```
Agent/IR set retryCount_ 0
Agent/IR set offset_ 0
Agent/IR set lastreceived 0
Agent/IR set lastpacket_ 0
```

The implementation of the proposed IR protocol without congestion delay option in IR.cc is shown in the following scripts and flowcharts.

Constructor:

```
IRAgent::IRAgent() : Agent(PT_IR)
{
	bind("packetSize_", &size_); //binding packetSize_ parameter in C++ and size_ in tcl.
}
IRAgent::IRAgent(packet_t type) : Agent(type)
{
	bind("packetSize_", &size_); //binding packetSize_ parameter in C++ and size_ in tcl.
}
```

The send method (void IRAgent::send(Packet* pkt, Handler *h)) and the receive method (void IRAgent::recv(Packet* pkt, Handler *h)) flowcharts are shown in Figure 1.

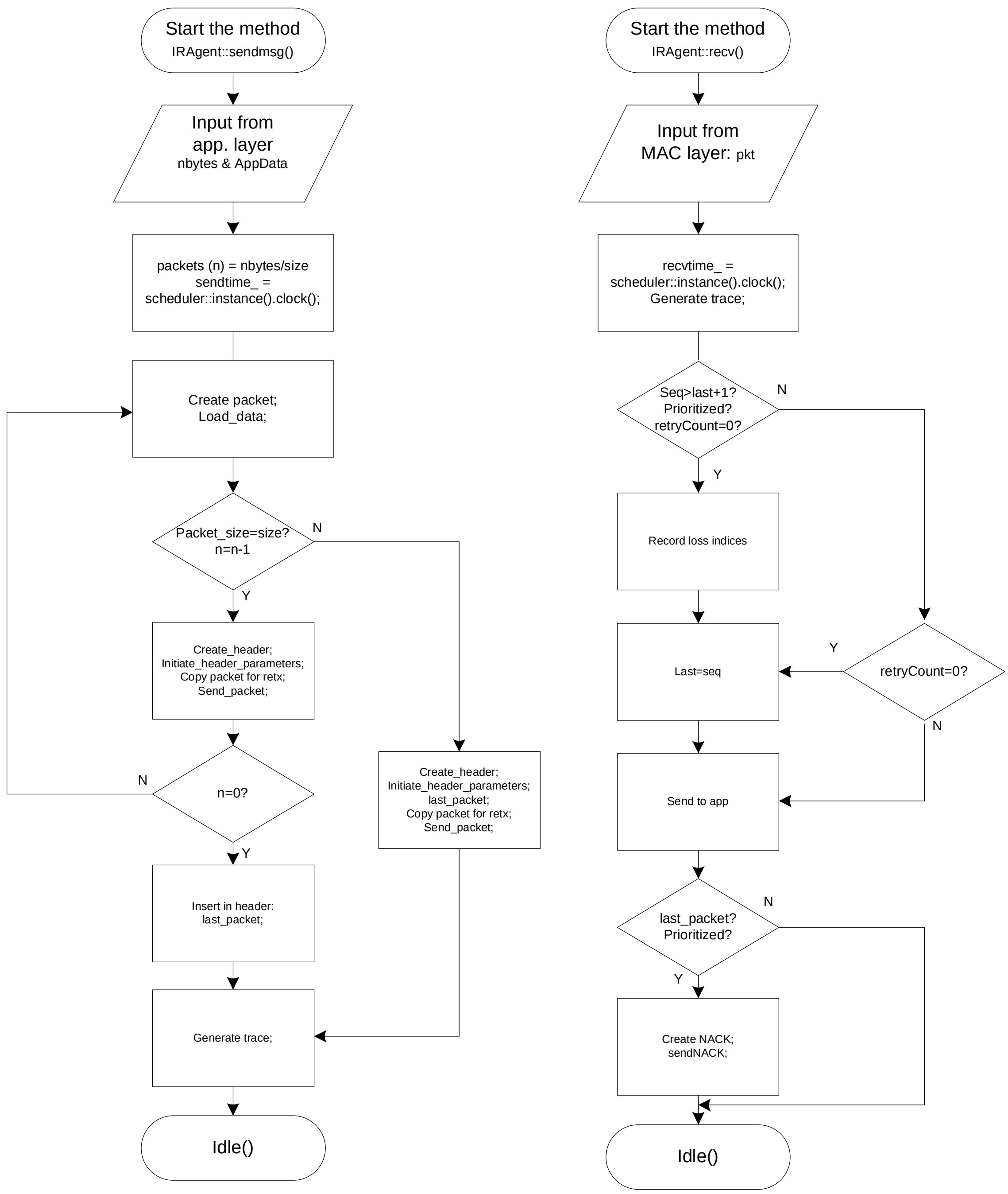


Figure 1: The sender and receiver methods of the proposed IR

2.2. Congestion delay

Congestion delay prioritizes sending the retransmitted packet rather than the regular one. Therefore, the protocols requires timer to postpone regular retransmission if lost packet is requested. Congestion delay value is determined by user in tcl script. When timer is activated, current packet transmission is postponed, and the retransmission is performed. Once timer is expired, regular packet transmission continuous. The following scripts show how to implement the timer. The timer is initiated in file IR.h.

```
class CDTimer : public TimerHandler {
public:
        CDTimer(IRAgent* t) : TimerHandler(), t_(t) {}
        inline virtual void         expire(Event*);
        protected: IRAgent* t_;
};
class IRAgent : public Agent {
public:
        friend class CDTimer; IRAgent(); IRAgent(packet_t); //inserted
protected:
        CDTimer cb_timer; //inserted
}
```

Constructor implementation in IR.cc:

```
IRAgent::IRAgent() : Agent(PT_IR), cb_timer(this)
{
        bind("packetSize_", &size_); //binding packetSize_ parameter in C++ and size_ in tcl.
}
IRAgent::IRAgent(packet_t type) : Agent(type), cb_timer(this)
{
        bind("packetSize_", &size_); //binding packetSize_ parameter in C++ and size_ in tcl.
}
```

The `CDTimer::expire` and The `execution_method` are added and in IR.cc:

```
void CDTimer::expire(Event*)
{
        //Executed method if timer is expired
        t->execution_method;
}
```

2.3. Transport layer redundancy

Transport layer redundancy uses a timer to transmit redundancy. The IR.cc script for redundant transmission is similar to previous codes, except the send method is duplicated as an execution method. Timer is set to transmit in 50% of inter-frame time. When timer is expired, the execution method is called and the redundant packet is transmitted. The redundant packet retryCount is set to 1, so the lost redundant packet is not retransmitted.

3. MAC layer implementation

3.1. Transport-MAC cross-layer protocol

The MAC cross-layer functional in the proposed protocol detects NACK packet sent by SSs to request retransmission. After detecting NACK packet, MAC layer reads how many packet requested by NACK packet. In order to do so, MAC layer module in WiMAX base station should be able to distinguish NACK packet from other type of packets. The following script is added in mac802_16SS.cc. This NACK detection will work if NACK packet creation in the sender adds parameter requested packet_ in IR.cc and IR.h.

```
void Mac802_16SS::receive ()
{
        assert (pktRx_);
        struct hdr_cmn *ch = HDR_CMN(pktRx_);
        if (ch->ptype()==PT_NACK)
        lost_=ch->requestedpacket_;
        .
        .
        .
}
```

and   in mac802_16SS.h

```
protected:
        int lost_;
```

After NACK request is detected, SS should add the number of the requested bytes in the nearest bandwidth request in ssscheduler.cc. The addition will increase the requested bandwidth to transport the retransmitted packet(s).

```
/**
 * Create a request for the given connection
 */
void SSscheduler::create_request (Connection *con)
{
        if (con->queueLength()==0)
                return; //queue is empty
        else if (mac_->getMap()->getUlSubframe()->getBw_req()->getRequest
                (con->get_cid())!=NULL) {
                debug2 ("At %f in Mac %d already pending requests for cid=%d\n",
                NOW, mac_->addr(), con->get_cid());
                return; //there is already a pending request
        }
        Int addBW = mac_->lost; //obtain NACK detection result
        Packet *p= mac_->getPacket();
        hdr_cmn* ch = HDR_CMN(p);
        bw_req_header_t *header = (bw_req_header_t *)&(HDR_MAC802_16(p)->header);
        header->ht=1;
        header->ec=1;
        header->type = 1;
        header->br = con->queueByteLength() + addBW*1076; //Adding additional bytes.
        header->cid = con->get_cid();
        .
        .
        mac_->lost=0; //Reset additional bandwidth to ensure increment occurs only once.
}
```

3.2. Bandwidth request mechanism

The proposed bandwidth request uses two request methods: reduced contention window for prioritized packets and piggyback method for non-prioritized packets. The first stage is to detectthe prioritized request and implement reduced window contention request for it. Contention window in contentionrequest.cc is modified.

```
ContentionRequest::ContentionRequest (ContentionSlot *s, Packet *p)
{
        assert (s);
        assert (p);
        s_=s;
        mac_ = s_->map_->getMac();
        window_ = s_->getBackoff_start();
        nb_retry_ = 0;
        p_=p;
        backoff_timer_ = new WimaxBackoffTimer (this, mac_);
        timeout_timer_ = new ContentionTimer (this);
        hdr_cmn* ch = HDR_CMN(p);
        if(ch->frametype_==1){//if contention is applied if packet is prioritized
                window_=0.5*window_;//contention window is reduced
        }
        .
        .
        .
}
```

Once SS receives bandwidth allocation, SS compares the allocated bandwidth to its buffer length. If data in buffer is greater than the allocated bandwidth, piggybacking is required. The script is implemented in transfer_packet method in file wimaxscheduler.cc.

```
int WimaxScheduler::transfer_packets (Connection *c, Burst *b, int b_data)
{
```

```
        Packet *p;
        hdr_cmn* ch;
        hdr_mac802_16 *wimaxHdr;
        double txtime, txtime2;
        int txtime_s;
        bool pkt_transfered = false;
        OFDMPhy *phy = mac_->getPhy();
        int bufferlength=c->queueByteLength(); //check buffer length.
         p = c->get_queue()->head();
        .
        .
        .
        if(max_data<HDR_MAC802_16_SIZE||(c->getFragmentationStatus()!=FRAG_NOFRAG
                &&max_data<HDR_MAC802_16_SIZE +    HDR_MAC802_16_FRAGSUB_SIZE)){
                ch->piggy_=bufferlength-b_data;// set piggybacked bytes.
                return b_data;
        }
        .
        .
        .
}
```

Besides video data, WiMAX also sends management data which is separated from video data. Therefore the following scripts are added.

```
if (b_data>manage.size){
        ch->piggy_=bufferlength-b_data+AvFrameSize;
}else{
        ch->piggy_=0;
}
```

If video data size is greater than manage.size, piggyback is performed and the predicted next frame size should be added. Otherwise, contention request is applied for the next data and scheduled by implementing the following script in file ssscheduler.cc.

```
/**
 * Schedule bursts/packets
 */
void SSscheduler::schedule ()
{
        .
        .
        .
        //check UL sub-frame to find the SS allocation (simplified code)
        for (int index = 0 ; index < map->getUlSubframe()->getNbPdu (); index++) {
                //check basic, primary and secondary connection
                b = map->getUlSubframe()->getPhyPdu (index)->getBurst (0);
                if (peer->getBasic(OUT_CONNECTION)!= NULL) {
                        b_data = mac_->getScheduler()->transfer_packets
                        (peer->getBasic(OUT_CONNECTION), b, b_data);
                }
                if (peer->getPrimary(OUT_CONNECTION)!= NULL){
                        b_data = mac_->getScheduler()->transfer_packets
                        (peer->getPrimary(OUT_CONNECTION), b, b_data);}
                if (peer->getSecondary(OUT_CONNECTION)!= NULL){
                        b_data = mac_->getScheduler()->transfer_packets
                        (peer->getSecondary(OUT_CONNECTION), b, b_data);}
                if (peer->getOutData()!=NULL){
                        b_data = mac_->getScheduler()->transfer_packets
                        (peer->getOutData(), b, b_data);}
                if(b_data<manage.size){
                        mac_->cr=0;
                }
```

```
        }
        .
        .
        .
}
```

After finding the allocated slot in UL and checking data sent in all connections, SS generates contention request for the next incoming data if mac_->cr=0. Contention request is also generated if the incoming data is higher than the average size of non-prioritized frame. The following script is applied in the same file.

```
if (mac_->y==0 || peer->getBasic(OUT_CONNECTION)->queueByteLength() > AvFrameSize)
        create_request (peer->getBasic(OUT_CONNECTION));}
if (mac_->y==0||peer->getPrimary(OUT_CONNECTION)->queueByteLength()> AvFrameSize)
        create_request (peer->getPrimary(OUT_CONNECTION));}
if (mac_->y==0||peer->getSecondary(OUT_CONNECTION)->queueByteLength()> AvFrameSize)
        create_request (peer->getSecondary(OUT_CONNECTION));}
if (mac_->y==0||peer->getOutData()->queueByteLength()> AvFrameSize)
        create_request (peer->getOutData());}
```

When receiving data from SS, BS checks either piggyback is requested. Therefore, a script is inserted in BS mac layer which is in file mac802_16BS.cc. If piggyback request exists, the requested bytes are added in existing connection.

```
/**
 * Process the fully received packet
 */
void Mac802_16BS::receive ()
{
        assert (pktRx_);
        struct hdr_cmn *ch = HDR_CMN(pktRx_);
        .
        .
```

```
        .
        //if piggyback!=0, allocate BW in current connection
        if(HDR_CMN(pktRx_)->piggy_>0){
                debug ("piggyback = %d\n", HDR_CMN(pktRx_)->piggy_);
                con->setBw(HDR_CMN(pktRx_)->piggy_ & 0x7FFFF );
        }
        .
        .
        .
}
```

3.3. Scheduling algorithms

The proposed schedulers are implemented by modifying schedule method in file bssscheduler.cc. The following is the implementation of round robin based scheduler.

```
/**
 * Schedule bursts/packets
 */
void BSScheduler::schedule ()
{
        .
        .
        .
        PeerNode *tp[10];  //Array to store non prioritized request
        int index =0;
        peer = mac_->getPeerNode (nextUL_);  //Retrieve peer from UL
        if (peer == NULL) //If the node is not here, starts from the beginning
                peer = mac_->getPeerNode_head();
        if (peer) { //if peer active, check all peers
                for (int i=0; i<mac_->getNbPeerNodes() && ulduration < maxulduration ;i++) {
                        if (peer->getReqBw()>0&&!getMac()->isPeerScanning
                                (peer->getAddr())) { //Check if peer has request
```

```
                if(peer->ft_!=1){ //If peer requests non-prioritized traffic
                tp[index]=peer; //then keep the request
                index++;
                }else{ //If it is the prioritized one, then process its request
                                .
                                . //Bandwidth allocation scripts
                                .
                        }
                }
        }
        if(i==mac_->getNbPeerNodes()-1 && index>0){ //If all peers have been check
        for(int k=0;k<index && ulduration < maxulduration;k++){
                        //lets serve non prioritized request
                peer=tp[k];
                .
                . //Bandwidth allocation scripts
                .
        }
        //Start from beginning
        peer = peer->next_entry();
        if (peer == NULL)
        peer = mac_->getPeerNode_head();
        if (peer)
        nextUL_ = peer->getAddr(); //go to next one
        else
        nextUL_ = -1; //will go the the beginning of the list
    }
}
```

FIFO based scheduler is also implemented in the same method and file.

```
PeerNode *tp[10]; //Array to store non prioritized request
```

```
int index=0;
bool prioritized = false;
int next = mac_->next; //Retrieve the first incoming peer
if(next!=0){
        peer = mac_->getPeerNode (next-1);
        mac_->next=0;
}else{
        peer = mac_->getPeerNode (nextUL_); //If no incoming peer, retrieve the next one
}
if (peer == NULL) //if the peer is not here, lets start from the beginning
        peer = mac_->getPeerNode_head();
if (peer) {
        for (int i=0; i<mac_->getNbPeerNodes() && ulduration < maxulduration ;i++) {
                if (peer->getReqBw()>0 && !getMac()->isPeerScanning (peer->getAddr())) {
                        //Store non prioritized request
                        if(peer->ft_!=1){
                                tp[index]=peer;
                                index++;
                        }else{
                                .
                                . //Bandwidth allocation scripts
                                .
                        }
                }
        }
        //Proses delayed request
        if((prioritized || i==mac_->getNbPeerNodes()-1) && index>0){
                for(int k=0;k<index && ulduration < maxulduration;k++){
                        peer = tp[k];
                        .
                        . //Bandwidth allocation scripts
```

```
                        .
                }
        }
        if(mac_->next!=0){
                peer = mac_->getPeerNode (mac_->next-1);
        }else{
                peer = peer->next_entry();
        }
        if (peer == NULL)
                peer = mac_->getPeerNode_head();
        if (peer)
                nextUL_ = peer->getAddr(); //go to next one
        else
                nextUL_ = -1; //will go the the beginning of the list
        }
}
```