Cross-layer mimo-links exploiting packet switching and adaptive modulation for TCP/IP enabled volcano monitoring networks

Erwin Anggadjaja

2014

Erwin Anggadjaja. (2014). Cross-layer mimo-links exploiting packet switching and adaptive modulation for TCP/IP enabled volcano monitoring networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/59974>

<https://doi.org/10.32657/10356/59974>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**CROSS-LAYER MIMO-LINKS
EXPLOITING PACKET SWITCHING
AND ADAPTIVE MODULATION FOR
TCP/IP ENABLED VOLCANO
MONITORING NETWORKS**

**ERWIN ANGGADJAJA
SCHOOL OF COMPUTER ENGINEERING
2014**

**CROSS-LAYER MIMO-LINKS
EXPLOITING PACKET SWITCHING
AND ADAPTIVE MODULATION FOR
TCP/IP ENABLED VOLCANO
MONITORING NETWORKS**

ERWIN ANGGADJAJA

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2014

Preface

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD) at the School of Computer Engineering, Nanyang Technological University (NTU).

The research reported in this thesis has been carried out in the period from August 2008 to December 2012. It includes the coursework requirements for various postgraduate subjects during the first two years, followed by extensive research as reported in this thesis.

During the time, I was supervised by Professor Ian Vince McLoughlin and co-supervised by Professor Lau Chiew Tong and Professor Annamalai Benjamin Premkumar.

Professor I. McLoughlin is now with the School of Information Science and Technology, University of Science and Technology of China.

Professor L. C. Tong is with the School of Computer Engineering, Nanyang Technological University, Singapore, and Professor A. B. Premkumar is now with Universiti Malaya, Kuala Lumpur, Malaysia.

Acknowledgement

Many thanks to God, in Jesus Christ, for the opportunity and strength to pursue this PhD, that I am able to progress much up to this point. Only by His abundant grace and unbelievable guidance have I been able to persevere, despite all the difficulties I have coped with.

Let me utter my deepest gratitude and warmest appreciation to the following people, who in many ways have contributed and inspired me during my PhD program which is starting from the extensive coursework, seminars, discussions, and seat-in exams, tough candidature process and assessment, and the most importantly work and research towards this thesis writing.

Foremost, I am truly indebted greatly to Professor Ian V. McLoughlin, my excellent and wonderful supervisor and mentor, for his vast knowledge regarding my research topic, his encouragement for myself and untiring guidance upon me. I am very grateful of his supportive deeds, kindness, and helpful advice. Also, I want to thank him for his trust and for bearing with me, particularly in many tough situations. No words could express my biggest appreciation to him.

Id like also give appreciation and gratitude to Professor Lau Chiew Tong and Professor A. B. Premkumar for his kind support at the end of my thesis writing. Their patience and beneficial assistance have contributed to the end of my PhD program.

I also express my thankfulness to Professor Kerry Sieh, Director of Earth Observatory of Singapore (EOS), for his continual financial support and kind cooperation during my settlement. Also Professor Chris Newhall, for his kindness and support

towards completing this thesis work.

My two best colleagues: Nguyen Quang Huy (Bryan) who patiently helped me adapt during my early years in NTU; Dr. Hamid Reza Sharifzadeh who kindly assisted and guided me concerning any research work.

Technical support executives in Parallel and Distributed Computing Centre (PDCC): Ms. Ng-Goh Siew Lai (Irene) for her help and support for providing resources and facilities in PDCC.

My friend in EOS: Phan Quoc Huy for sharing insightful idea and his assistance during my research; Lee Gek Leng (Janice) for her patience, generous and helpful support; and largely for May Pea Jingmei for her kindly help and bighearted approach in assisting me.

Last but not least, my greatest love for my family in Jakarta: My parents, for their constant support and for always believe in me; My brother and his lovely family; and the most for my beloved wife Nia for her endless prayers and encouragement, and for always brightening my days while I am in NTU.

Contents

Preface	i
Acknowledgement	ii
List of Figures	viii
List of Tables	xi
List of Abbreviation	xii
Abstract	xiii
1 Introduction	1
1.1 Overview	1
1.1.1 Future wireless services	1
1.1.2 Wireless service network for Earth Observation	3
1.2 Motivation	4
1.3 List of publication	6
1.4 Organisation of Thesis	7
2 Network simulation tools and OMNeT++	10
2.1 Wireless Network and Modelling	10
2.2 Network simulation	12
2.2.1 Cautions and motives behind the use of simulators	13
2.3 Comparisons of several network simulators	16
2.3.1 ns-2	16
2.3.2 OPNET Modeler	18
2.3.3 OMNeT++	20
2.4 Modelling in OMNeT++	23

2.4.1	INET framework model	24
2.5	OMNeT++ Simulation: Lost and Received Packets	24
2.5.1	Concept	24
2.5.2	Bit and Packet Error Rate	25
2.5.3	Experimental scenario	26
2.5.4	Results and discussion	27
2.6	Summary	29
3	Segmentation and Reassembly with Error Control mechanism	31
3.1	Overview	31
3.2	Segmentation and Reassembly	31
3.3	Error control mechanism: Retries and ReACKs	32
3.4	Bits transmission efficiency	34
3.5	Experimental scenario: Realistic wireless transmission with error control	34
3.6	Results and discussions	36
3.7	Summary	40
4	TCP-Based Transmission Model	41
4.1	Overview	41
4.2	Classical wired vs. wireless protocol	41
4.3	Standard transport protocol	42
4.3.1	TCP	44
4.4	TCP over wireless links	45
4.4.1	Drawbacks	46
4.4.2	Improvements	46
4.4.3	Remarks	50
4.5	TCP-Based Point-to-point Transmission Model with SAR and Error Control	50
4.5.1	Segmentation and Reassembly	52
4.5.2	Error Control	53
4.5.3	Evaluation methods	54
4.5.4	Discussions	56
4.6	Summary	61

5	Multi Parallel Independent Link Transmission Model Implementing Switching and Retries Scenario for Different Channel Conditions	63
5.1	Overview	63
5.2	Diversity and MIMO technology	64
5.2.1	Diversity	64
5.2.2	MIMO technology	67
5.3	Dual- Parallel Independent Link Model	70
5.3.1	Channel difference	71
5.3.2	Switching methods	73
5.3.3	Analysis of switching methods and channel difference model . .	74
5.3.4	Retries scenario	76
5.3.5	Analysis of switching methods and retries scenario under different channel conditions model	78
5.3.6	Another possible channel scenario	80
5.4	Quad- Parallel Independent Link Model	82
5.4.1	Altering-Link Modelling	82
5.4.2	Discussion-1	84
5.4.3	Discussion-2	86
5.4.4	Discussion-3: Randomising channel's BER	87
5.5	System Complexity	89
5.5.1	Computational and Deployment Complexity	89
5.5.2	Latency and Buffer requirements	90
5.6	Summary	91
6	Quad-PILM with Adaptive Modulation	92
6.1	Overview	92
6.2	Adaptive Modulation	93
6.2.1	Previous work on AM	94
6.3	Modelling	96
6.4	Quad-PILM without Adaptive Modulation	97
6.5	Quad-PILM with Adaptive Modulation	98
6.5.1	AM-scenario	98
6.5.2	AM-scenario with link value parameter	100
6.5.3	Updated link value AM-scenario	102
6.5.4	Summary	105

7 Conclusion and Future Work	108
7.1 Conclusion	108
7.2 Future Work	113
Appendices	115
A Multi-point network	117

List of Figures

2.1	OMNeT++ Model Structure	23
2.2	Simulation topology.	26
2.3	Lost packets vs. BER, for $N = 1024$ & 2048 bits.	27
2.4	Sent packets vs. BER, for $N = 1024$ & 2048 bits.	28
3.1	Concept of segmented data and block size	32
3.2	Error control mechanism: re-ACK scenario once the timeout expires . .	33
3.3	Nodes A & B flowchart	35
3.4	PER vs. BER of various data sets, with $w = 49$	36
3.5	PER vs. BER of various data sets, with $w = 196$	36
3.6	Total Retries and ReACKs vs. BER, with $w = 49$	38
3.7	Total Retries and ReACKs vs. BER, with $w = 196$	38
4.1	TCP/IP protocol suite: TCP and UDP	43
4.2	TCP connection and establishment.	47
4.3	Layer and data handling structure of single-link TCP-based point-to-point model	51
4.4	TCP-based point-to-point transmission model.	51
4.5	Encapsulation and Decapsulation concept.	52
4.6	Decapsulation and Segmentation process implementation.	52
4.7	Concept of sequence number and segment size, and relationship to TCP/IP frame and original file.	53
4.8	Concept of maximum retry mechanism, r_{MAX}	54
4.9	n separate segment transmissions, each containing N bytes of data, are made. Each segment includes a one byte sequence number. Due to errors, some segments need to be re-transmitted, maybe more than once. So m segment retransmissions will be necessary to successfully convey all data.	55

4.10	Goodput comparisons for different segment size and maximum number of retransmission.	57
4.11	Average <i>goodput</i> vs. <i>SS</i>	60
5.1	Receive and Transmit Diversity for the simplest scenario, where the figure shows SIMO (Single-Input Multiple-Output) 1×2 and MISO (Multiple-Input Single-Output 2×1 arrangements respectively.	66
5.2	Diagram of MIMO (Multiple-Input Multiple-Output) wireless transmission system.	68
5.3	Switching methods illustrated with two TCP payloads each divided into four frames and transmitted on successive timeslots over two physical links.	74
5.4	Dual-link model with different switching methods and channel difference.	75
5.5	Two retry scenario for the dual-link transmission model for both SWS and PLD switching mechanisms.	77
5.6	Comparison of all dual-link performance methods for different channel conditions, $BER = 10^{-5}$	79
5.7	Comparison of all dual-link performance methods for different channel conditions, $BER = 10^{-4}$	79
5.8	Link A consistently get worse, $BER = 10^{-5}$	80
5.9	Link A consistently get worse, $BER = 10^{-4}$	81
5.10	Quad-link switching methods and retries scenarios showing (i) S-RDL, (ii) P-RDL, (iii) S-RRL, (iv) P-RRL	83
5.11	Quad-PILM: SWS-switching of all retry mechanisms for $BER = 10^{-3}$	84
5.12	Quad-PILM: PLD-switching of all retry mechanisms for $BER = 10^{-3}$	85
5.13	Quad-PILM altering link: SWS-switching of all retry mechanisms for $BER = 10^{-3}$	86
5.14	Quad-PILM altering link: PLD-switching of all retry mechanisms for $BER = 10^{-3}$	87
5.15	Randomized channel BER.	88
6.1	Quad-PILM with no-AM, $r_{MAX} = 4\times$ and $6\times$	98
6.2	The Quad-PILM system with AM triggers a modulation adjustment when m_{NAK} erroneous resends have occurred.	99
6.3	AM scenario with LV parameter– with m_{NAK} is counted for each link (left) and collectively at node base (right).	100
6.4	No-AM versus AM-LV ₁ with varying m_{NAK}	101
6.5	No-AM vs. AM-LV ₂ scenario, with varying m_{NAK}	103
6.6	AM-LV ₂ scheme for channel distribution between 0.8 and 0.9.	105

6.7	AM-LV ₂ scheme for channel distribution between 1.2 and 1.3.	106
6.8	AM-LV ₂ scheme with different m_{NAK} value.	106
7.1	A typical multi-sensor volcano monitoring network deployment [1]. . . .	114
A.1	Multi-point network scenario.	118
A.2	Goodput and Finishing time of Path-2, BER = 10^{-5}	119
A.3	Goodput and Finishing time of Path-2, BER = 10^{-4}	120
A.4	Goodput and Finishing time of Path-3, BER = 10^{-5}	122
A.5	Goodput and Finishing time of Path-3, BER = 10^{-4}	123

List of Tables

2.1	Simulation-1 parameters	27
2.2	Network simulation tools comparison	30
3.1	Simulation-2 parameters	35
3.2	Comparison of PER & total bits sent on 3 retries, BER= 10^{-3} and $w=49$.	37
3.3	Comparison of PER & total bits sent on 5 retries, BER= 10^{-3} and $w=49$.	37
3.4	Comparison of PER & total bits sent on 3 retries, BER= 10^{-3} and $w=196$.	39
3.5	Comparison of PER & total bits sent on 5 retries, BER= 10^{-3} and $w=196$.	39
4.1	Single-link simulation parameters	56
5.1	Dual-link B_T distribution	73
5.2	Quad-link B_T distribution	83
6.1	BER value for different QAM modulation at SNR = 15 dB, as reported in [2].	97

List of Abbreviation

Abbreviations used in this thesis:

ARQ (automatic repeat request)
AM (adaptive modulation)
AM-LV (adaptive modulation - link value)
BER (bit error rate)
bps (bit per second)
Kbps (kilobit per second)
Mbps (megabit per second)
MIMO (multiple-input multiple-output)
PER (packet error rate)
PLD (payload switching)
PRDL (payload switching - retries to different link)
P-RRL (payload switching - retries to random link)
SNR (signal to noise ratio)
SAR (segmentation and reassembly)
SISO (single-input single-output)
SWS (segmented window switching)
S-RDL (segmented switching - retries to different link)
S-RRL (segmented switching - retries to random link)
TCP (transport control protocol)
UDP (user datagram protocol)
QAM (quadrature amplitude modulation)
Q-PILM (quad-link parallel independent link model)

Abstract

The effects of sub-channel packet routing and adaptive modulation (AM) are explored in this thesis¹ for MIMO linking system under differing channel conditions. The challenges is to exploit time-varying imbalances between logical sub-channels by different cross-layer routing and modulation strategies to improve overall goodput.

An investigation of several network simulation tools is presented in order to choose an appropriate simulation and modelling tool, particularly to be used and fitted to the wider EOS project. Several important aspects such the availability of protocol model, simulation capabilities and academic acceptance, emphasis on unsophisticated attributes such as debugging and GUI environment, open source nature, and cost issues are also considered to accommodate non-specialist users (i.e. earth scientists) who are the target recipients of the wider EOS research project. Discussions and analysis are given prior to selecting OMNeT++ as the basis for the ongoing simulations. Further assessments were made and verified by presenting a simulated model implementing segmentation and reassembly (SAR) concepts combined with local error control mechanism in the form of ACK/NACK triggered retries with timeout.

As a main contribution, MIMO sub-channel characteristics are modelled in terms of frame-by-frame BER, channel delay, and data rate to explore the effect of channel BER imbalance by implementing frame-by-frame sub-channel re-routing and retry

¹This research is part of a wider project within the Earth Observatory of Singapore (EOS) concerned with future earth observation networks, particularly in the field of volcano monitoring networks.

mechanism with the aim of improving overall goodput. Several simulations are derived which make use of the TCP protocol through the use of an open-source simulation package within the OMNeT++ environment named INET. Scenario models are created from a single protocol link and then extended into dual wireless sub-channels, and the final interesting quad- parallel independent link model (Quad-PILM). For both dual- and quad- link models, two novel switching methods SWS and PLD and two retry scenarios are set up: RDL and RRL, to explore how re-transmissions can be adjusted to take advantage of short-term channel imbalances. We investigate the system in the light of goodput under various channel differences.

As a final model, a BER-directed adaptive modulation switching scheme, based on a quad- parallel link model, is evaluated. This cross-layer design relates to physical, data link, and transport layers. Although it technically breaks a strict layering concept, even a simple switching scheme with a very clean layered interface, is shown to yield a performance improvement. Based on our analysis, we determine the degree of improvement achievable through the proposed scheme, and show that the technique may be advantageous for multiple antenna systems employed in cross-layer designs.

This thesis demonstrates that even quite simple switching heuristics and two-way packet re-routing mechanisms can provide an overall improvement in a common real-world situation where MIMO sub-channels exhibit slightly unequal error rates for durations of one or more transmission frames.

Chapter 1

Introduction

1.1 Overview

1.1.1 Future wireless services

In recent years, the substantial and rapid growth in the wireless communication industry has gained momentum for many types of wireless services. Combined with the progress of the Internet, it is apparent that there will be increasing demand for wireless data services [3]. Future wireless traffic is anticipated as a mix of real-time traffic (including voice, multimedia, and data traffic such as file transfers, web browsing, etc), thus the technologies that power these services must be able to deal with many challenges while satisfying the demand for capacity, high data rates, and good quality of service (QoS) (for example, in terms of latency and error rate). Besides these issues, there is an ongoing need to maximize throughput/goodput performance through efficient and effective spectral utilisation.

Many mechanisms have been advised to support data traffic over wireless media – from wireless local area networks (WLANs) –based on the IEEE 802.11b– to wireless wide area networks (WWANs) where data services are supported in the 2.5- (2.5G)

and third-generation (3G) cellular systems. Unfortunately wireless transmission can be unreliable for service delivery [4], leading to poor user experience. The overall stability of a wireless system results from many factors in wireless communication such as multipath fading, path-loss and effects of reflection, shadowing, diffraction, and scattering, all of which are statistically unstable [5]. Not mentioning some undesired obstructions, causing loss of LOS (line of sight), or interference caused by extreme weather conditions.

There are several methods worthy of investigation to improve wireless network performance. In terms of wired networks, TCP/IP has become the preeminent internet data transport protocol, precisely because it leads to reliable and stable transmission over potentially unreliable links. It is also widely interoperable and extremely pragmatic in its operation. However, when deployed similarly in a wireless context, TCP/IP suffers from well-known performance issues. Nevertheless, much research has been conducted on exploring TCP performance over wireless networks, and many solutions have been proposed, for example including lower-layer enhancement, cross-layer design approaches, as well as new mechanisms for improving the underlying transport protocol. These work will be elaborated later in the thesis.

Within the wireless domain, there are two further techniques that are worthy of mention and relevant to this thesis. The first is multiple-input multiple-output (MIMO) wireless communication. These MIMO systems have become well-known for their ability to overcome harmful effects of fading through the use of multiple antennas at both ends of communication systems. MIMO can enhance data rate without necessarily increasing bandwidth or transmission power [6]. Thus MIMO technology has become popular because of its ability to provide higher spectral efficiencies and greater throughput or goodput to the end user. For this reason, it is considered as a foundational method for present and future wireless systems.

The second technique, Adaptive Modulation (AM), is a useful and practical

method of managing wireless link error and bit rate tradeoffs in deployed systems. This technique adapts modulation and/or transmission power according to the channel propagation conditions, interference scenarios, and traffic or data rate requirements. It usually involves a switching mechanism rather than a continuous adaptation, with its aim being to enhance the spectral efficiency of wireless systems while assuring QoS requirements. AM has been investigated widely, with the purpose of matching transmission rates to varying channel conditions. In relation to the various implementation layers, AM at the physical layer is commonly designed independently from other layers. But, this premise may not be either practical or optimal in all situations. This has been analyzed by some authors, particularly in regard to cross-layer design solutions. More explanation, and background information concerning AM can be found in Chapter 6.

1.1.2 Wireless service network for Earth Observation

The motivation for this research is in the deployment of future earth observation networks¹. The challenges of wireless sensor network deployment in this area have been studied and published in several cases, for example: [8] [9] [10] [11] [12] and [13]. In particular, Werner-Allen et. al. [14] suggested that wireless sensor networks have vast potential for monitoring seismic events such as volcanoes, though some challenges still remain, for example: number of channels, communication bandwidth, power requirements, storage, high sampling rate, etc.

In general, conventional seismic monitoring node(s) that base their monitoring systems on traditional wireless sensor networks (WSNs) have limitations, especially in terms of bandwidth and data reliability for analyzing and predicting future seismic events. However, they have been shown to be particularly useful for developing

¹We limit the implementation justification to volcano monitoring as this research is funded by EOS [7] under the volcanology discipline, however it is equally applicable to other multiple channel TCP-enabled sensor networks.

applications in areas of volcanic or seismic activity [13]. The wireless links, and in particular the mesh configuration, permit those nodes to communicate with each other and reliably deliver data to a base station through a sink-node, possibly using several hop-to-hop routes. Despite this great potential, these networks are costly to deploy, complex to design (requiring WSN specialists) and have been consistently unreliable in operation. Within EOS, the approach considered for future networks is to use plug-and-play TCP/IP enabled high-bandwidth links: this allows sensors to be independent of the underlying wireless technology (as long as both support TCP/IP), not requiring specialist skills to set up, and able to easily adopt the latest high-bandwidth wireless technology.

1.2 Motivation

Building upon the premises described above, and the observation that reliable wireless links for TCP/IP increasingly adopt MIMO technology, this thesis investigates and proposes adaptations to MIMO TCP/IP links that may be useful to the eventual realistic deployment of such systems. As a requirement of the research funding, the work should potentially benefit future TCP/IP earth observation networks, as well as build a simulation and modelling baseline suitable for such networks (i.e. the OMNet++ model which is to be adopted by other EOS researchers). However the main contribution of this research is conducted at a much lower level – it involves adjustments to lower protocol layers, particularly cross-layer enhancements.

Thus, initially, a reliable simulation tool is required to model and predict the behaviour of real networks, and yet be easily accessed by non-specialists end users (such as earth scientists). The model should utilise TCP/IP as the main transport mechanism, and be adaptable to be able to incorporate link-layer error mitigation techniques to overcome the weaknesses of TCP/IP in a wireless environment. In

particular, we will investigate the effects of cross-layer packet switching mechanisms on the higher-level performance of the model, both with and without the use of an AM technique.

Following the model selection and construction, the main hypothesis of this thesis is based upon the observation that in deployed multichannel links, it is rare that instantaneous BER/SNR of each channel² is exactly identical [15], even with co-located polarized antennas. In practice, not only will one or more channel average BERs be slightly better or worse than others (perhaps due to antenna placement or electronics tolerances), but all channel BERs fluctuate with complex fading patterns at a rate that is normally significantly slower than the packet length or retraining time of the system [16] – except in the very unusual case when the system is continually operating under absolute worst-case rated conditions or beyond.

The premise is that a cross-layer design (CLD) solution, employing computationally simple rule-based packet (re-)routing over alternative channels, can enhance the goodput performance in realistic scenarios without additional transmission power or bandwidth. We will later extend this to considering a modulation switching concept. We specifically look for enhancement in user-experienced goodput, and will consider single, dual- and quad-link systems.

It is always worthy to compile recent research interest, specifically in the wireless environment. In relation to this, some interesting findings of our proposed model will also be discussed that may influence future system design. As can be seen later in this thesis, this would be beneficial for multiple antenna systems arrangement which employ a cross-layer design.

²In this context the word ‘channel’ refers to either logical or physical raw transmission path alternatives that are available at the physical layer of many MIMO systems.

1.3 List of publication

To support this thesis, the following are several work and publications made during the candidature period:

Journal paper

E. Anggadajaja, and I. McLoughlin, "Packet Switching, AM Adjustment and Retry Mechanisms for Cross-Layer MIMO Link Design", *Telecommunication Systems Journal*, (Accepted for publication).

Conferences

E. Anggadajaja, and I. McLoughlin, "Cross-layer MIMO-Links Exploiting Packet Re-routing Mechanisms and Adaptive Modulation in Diverse Channel Condition", *Australasian Telecommunication Networks & Application Conference (ATNAC) 2013*, Christchurch, New Zealand, 20-22 November 2013.

E. Anggadajaja, I. McLoughlin, and A. B. Premkumar, "TCP-based Multi Parallel Independent Link Model Implementing Packet Switching and Retries for Different Channel Conditions", *19th European Wireless Conference (EW 2013): 5G technologies and Spectrum Crunch in 2020*, Guildford, UK, 16-18th April 2013.

E. Anggadajaja and I. McLoughlin, "Point-to-point OMNeT++ based simulation of reliable transmission using realistic segmentation and reassembly with error control", in *IEEE Conference Publications, 2nd International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT)*, December 2010.

Book Chapter

E. Anggadajaja and I. V. McLoughlin. "Modelling WSNs using OMNeT++", in

Technological Advancements and Applications in Mobile Ad-Hoc Networks: Research Trends, Edited by K. I. Lakhtaria, IGI Global, 2012.

1.4 Organisation of Thesis

The structure and arrangement of this thesis is as follows:

Chapter 1 introduces an overview of this research regarding future wireless systems and the possible future deployment in the area of earth observation. It then follows by exploring the motivation behind this thesis. Then the contribution will be explained and related to the successive chapters.

In **Chapter 2**, several related works concerning simulation tools are reviewed. This is indicated as an initial, yet important, stage in this research, i.e. to choose and make careful decision of which simulation tool that is convenience to use in evaluating the models proposed later. This start with elaborating both advantages and disadvantages of using simulation tools, then exploring the features of the three most commonly used network simulator tools, which are ns-2, OPNET, and OMNeT++. Various related issues (i.e. availability of protocol and topology models, debugging and GUI environment, simulation comparisons, and open source & cost issues) are explored thoroughly to conclude by selecting OMNeT++ for the evaluation. Simulation and discussion of simple wireless transmission systems are developed to show and prove the performance of OMNeT++.

Progressing to a more advanced and realistic simulation model, **Chapter 3** will introduce segmentation and reassembly (SAR), a concept of dividing and re-accumulating packets, as this allows them to be safely transferred across error-prone transmission links. In addition to this, a combined error control mechanism (inspired by TCP) with retries and reACKs similar to the radio link protocol (RLP) is also added, considering overall system efficiency issues. These aspects will be explored

carefully, as the techniques will be adopted for the remainder of this thesis, and thus serves as a foundation. The simulation is carried out to compare the results against theory, with various trade-off issues also explored.

In **Chapter 4**, TCP/IP protocol support is adopted in extension of the basic simulation model. It makes use of the INET framework [17], an open-source network simulation package for the OMNeT++ environment. This contains models for both wired and wireless networking protocols, including TCP/ IP and 802.11 models. Two standard transport protocols, UDP and TCP are briefly explored, and possibility of using TCP over wireless links is discussed prior to modelling the wireless transmission model combined with SAR and error control mechanisms.

In **Chapter 5**, the TCP-based model is expanded with a MIMO-like structure, starting from single-link, then dual-link, and then lastly a quad-Parallel Independent Link Model (PILM). As novel ideas, two switching methods are presented named Segmented Window Switching (SWS) and Payload (PLD)-switching. Basically, these direct how the data packets are re-routed to another link once the original goes ‘bad’ and another is slightly ‘better’. During the simulation, their average BERs remain as specified.

In addition, to deepen the simulation model to be more realistic, two different retry rules are implemented. The first one, in which retries are re-sent to a different link (RDL), and the second exploiting a random link (RRL). The retry scenarios here are related to the description that is described in Chapter 3. The simulation model is evaluated specifically to look for an improvement in goodput, and investigated in the light of diverse error rates between those links.

In **Chapter 6**, adaptive modulation concepts are addressed in relation to the Quad-PILM model which is based on the combined PLD-switching method and RDL-based retry scenario. The idea is derived from the different BER values (reported by [2]) experienced by various QAM modulations over a fixed link. In this simulation,

once a channel suffers persistent errors, it will adjust its modulation scheme, i.e. both BER and data rate for that link will change. The simulations explore several modulation switching and retries scenarios, which lead naturally to a cross-layer design (CLD) solution. Some interesting findings will also be reviewed and discussed that may influence system design.

Lastly, **Chapter 7** will sum up the contributions made in this thesis and discuss in brief the future directions of research relating to these areas of research.

Chapter 2

Network simulation tools and OMNeT++

This chapter will present an overview of network simulation tools and an evaluation of their suitability for this research. Some discussions on the benefits and challenges will also be explored. We will concentrate on what are probably the three most common simulation tools available today, namely ns-2, OPNET, and OMNeT++, for which various related issues will be identified and examined. We will argue that the latter tool holds certain advantages for our research, despite the fact that it has not yet been widely adopted. To validate the use of OMNeT++, we develop a simple wireless transmission model for it. We then derive a theoretical analysis with which to compare our model and show that discrepancies between the model simulation and theory are minor.

2.1 Wireless Network and Modelling

Wired networks have long been known for their reliability. Today, they tend to be certain, secure, and fast in terms of speed. Nevertheless, they can be difficult to

manage and maintain for one main reason: the complicated wiring and connectivity required for the physical network connections. This inflexibility, along with cost issues, range and convenience factors have caused wireless networks to be preferred for many applications including, with recent developments of wireless technologies, for backbone networks.

Nowadays, wireless network adoption growth sees it becoming critical infrastructure in areas such as typical office environments, for government services, as well as industry. The mobile phone has become a common tool for conversation and convenience. Access points are now used for internet access instead of stand-alone routers and cables. Video conferencing is becoming more common for organization purpose (for example meetings, teleconferences, etc.), and even the use of GPS and smart phones for live broadcasting / news are now mainstream.

Developing network models for this kind of connectivity can be troublesome, costly and difficult, i.e. in terms of network size, timing as well as repeatability. Modelling then has become essential for designing, understanding and evaluating such networks [18]. Existing (and future) network technology can be improved using simulations. Protocols and algorithms can be simulated, tested, designed, and adjusted to alleviate the complexity of networks or to improve their performance based on almost any important criteria.

In the process of creating such models, the following questions may arise: (1) Can modelling such systems to some degree of accuracy ensure high reliability in the field (with some trade-offs accepted)? And (2) As simulated systems become larger and more complex (and deployment becomes more costly), a greater assurance is required that a system will work before it is deployed. Thus to answer these issues, we argue that modelling –in a sense of having an appropriate simulation tool– is necessary to resolve both issues. Furthermore, the simulation tools should be usable by those who

wish to deploy such a network¹.

2.2 Network simulation

A network simulator is definitely one of the major evaluation methodologies to represent, replicate, or analyze a network (with the network itself is not necessarily actually or physically present). Typically, simulation tools test protocols being developed, or perhaps are used to weigh options against competing communication architectures or protocols, when correctly modelled and specified the relevant operating behaviour of subcomponents (such as network nodes, channel links, etc). The aim is obvious, to model and predict the behaviour of the real systems as precisely as possible to predict behaviour of networks (throughput, delay, round-trip time, etc) with varied scenarios (bandwidth, file size, node(s) distance, simulation time).

Most up-to-date simulators are provided with support for the well-known protocols (Ethernet, TCP/IP, SCTP, IPv6, etc - depending on the simulator itself). Even several simulator frameworks have been created for specific network purposes - concentrated on specific layers with detailed models. For example, Castalia [19] (for Wireless Sensor Networks (WSN), Body Area Networks (BAN) and low-power embedded devices network), MiXiM [20] (another framework for mobile and fixed wireless networks for models of radio wave propagation, interference, power consumption and wireless MAC protocols (e.g. Zigbee), and INET (will be discussed later).

More models will be explained further in this chapter, however it is important to note that we restrict the discussion to the smaller, less expensive research tools, rather than the more professional, capable, and higher cost tools (for example Qualnet²). The reason is that the research discussed in this thesis was funded on the basis of advancing

¹For the current research, the eventual end-users are earth scientists, and thus the tool should be accessible to and modifiable by non-specialists.

²Formerly known as GloMoSim, it is a commercial simulator for both wired and wireless network by Scalable Network Technologies [21].

towards an eventual endpoint of a low cost, preferably open source, yet capable and fast tool for simulating large-scale scientific sensing networks. This eventual endpoint is not the goal of the current work (it will be the goal of one of several subsequent research projects), but it has strongly influenced the selection of the network simulator used in this research.

2.2.1 Cautions and motives behind the use of simulators

We take account several cautions when selecting a network simulator:

- a. Uniqueness and drawbacks of each simulator.

Some simulators are little more than ongoing development projects and thus updates are expected, i.e. they are active research communities and projects in themselves. A simplified or inaccurate modelling of library functions might therefore be the shortcoming of some immature simulation tools in this category. Moreover, not all simulators have relevant or well-known protocols included that would be needed to develop real-world scenarios.

- b. Uncertain results.

There also exists a danger when using simulation tools in that, the simulation results could be uncertain, and even some critical details may be missing. This is a particular risk with higher-level tools that may hide unrealistic models within a pretty GUI. A good judgement of validation could be use to verify the results, for example by using theoretical or mathematical analysis. It has been advocated by [22] that in such situations, the user should not blindly trust the simulation output – instead double-check the result or include a validation mechanism.

Even better of course is to use measurements or mathematical analysis³.

³Our simulation at the end of this chapter is designed for precisely this purpose: to demonstrate with a known baseline system that we yield a similar outcome and to compare our simulation results with the results of the analytical expressions.

c. Level of detail.

In relation to this, one must be aware that insufficient detail leads to coarse simulation results which may miss important information, whereas too much detail will make the simulation slow and complex [23]. In such case, for example in the general wireless network simulation, very complex models are usually not necessary needed – this could be counter-productive and time consuming though. Generally, the optimum degree of detail depends on the purpose and the desired nature of the study itself [18], but in many cases, we are interested to adjust our model to the right level⁴.

On the other hand, there are also a number of motives behind the use of simulators, such as:

- a. Many researchers publish their papers (i.e. present results) based on experimental simulation.

This type of investigation is both accepted and in some cases preferred in the research literature. This stems from the requirement for a reliable model to foresee the performance in a real world environment, and particularly for multifaceted high-level data transmission protocols. This is usually well suited to tackling large numbers of nodes (scalability issues) and particularly for systems with costly sensors and difficulties in real-world deployment.

- b. Practical implementation.

When considering to test a large-scale network, deploying sensor nodes in the field can be a time consuming and difficult task. Protocols, architecture, schemes, etc can be assessed in a very large scale easier by configuring defined parameters in a simulation.

⁴In this thesis report, we focus on the detail of the TCP-level framework (i.e. connection establishment, encapsulation/decapsulation model, connection termination, etc.), and also point out most of the assumptions used, as will be described later in the text. This follows and complies with the approach suggested by Floyd and Paxson [22].

c. Investment wise.

In terms of cost and time, building a network and real experiments can be very costly and always time consuming. Not mentioning the doubtful repeatability of tests using such systems. Simulation-based testing can help to suggest whether or not these time and monetary investments are wise, before undertaking a real-world deployment.

Therefore, we are advocates of the use of simulation and modelling as the most useful and common approaches to developing and testing new protocols for sensor networks. Thus we consider a number of network simulators related to sensor networks, as mentioned by [24]. All of them present both positive and negative features.

It is important, in order to effectively develop any protocols with the help of simulation, to know the benefits and weakness of the various simulators, and to aid us in choosing which simulators should be used in terms of flexibility, reliability, and budgetary constraints. For example, in terms of flexibility, a simulator should allow the user to modify their own model with little restriction. A commercial simulator may not meet the need for low-cost simulation –open source software might be a good decision in this respect, if it is shown to be accurate and sufficiently fast. Reliability is obviously an important factor in deciding which simulation tool to use⁵. Importantly, both accuracy and appropriateness of the model and tool are very important aspects in the chosen network simulator.

Several simulation frameworks have been widely used in telecommunication and networking research, for example ns-2, OPNET, GloMoSim, J-Sim, OMNeT++, etc, with most of them are based on the paradigm of discrete event simulation (DES)⁶

⁵More of these tools are categorized in [18], in which usability, execution speed, scalability, model diversity, fidelity ranges, and level of support are all classified.

⁶DES [25] [26] is common now in wireless network simulation. Some of its fine features are it can simulate jobs on different nodes simultaneously, making use of dynamic memory management in which entities can be added and dropped easily, and can list pending or completed events. Users can check and debug the code in a step-by-step manner without disturbing the running operation. In short, the simulated network is triggered when a message or a packet arrives at a particular node, then this successively processes more events in turn (queue operations, for example).

2.3 Comparisons of several network simulators

This section will be exploring about three mainstream simulators that may be suitable for our simulation needs which are ns-2, OPNET, and OMNeT++. We take account that vital factors influencing the choose of simulator are ease of use, cost, and importantly, the availability of simulation models for desired applications [27]. All simulators here come along with different environments and protocols, thus it is beneficial to contrast their differences for enhanced understanding. Comparisons are also discussed later in this document.

2.3.1 ns-2

Network Simulator, *ns*, developed by the Network Research Group at the Lawrence Berkeley National Laboratory (LBNL) as a variant of the REAL network simulator [28] in 1989 and then has evolved significantly over the subsequent few years. At present, it is popularly called *ns-2* [29], and has been extended to be possible for modelling wireless 802.11 and MANET (Mobile Ad-hoc NETwork) protocols. It maintains full backward compatibility for almost all functionality in version-1, as it establishes more features for configuration and programming.

The core of ns-2 is based on C++ due to its object-oriented features. It provides the interface through OTcl, an object oriented dialect of the Tcl scripting language. Thus ns-2 would be a compromising interaction between performance and user-friendly design. This is why it is very popular in research circles; in addition it is published as an open source model with online documentation.

ns-2 is one of many event based simulators. It is mainly used for ad-hoc networking research as well as simulation of network protocols, both wired and wireless - and has been claimed as one of the most popular network simulation tools for sensor networks. An object-oriented discrete event simulator, it has a range of extensions of

protocols already modelled, and allows users to create and write any new protocols. The architecture follows closely the OSI 7-Layer model. This extensibility feature possibly made it so popular for many developers to use, with its combination of ease and popularity and high number of different protocols built in.

In spite of this, ns-2 possesses some drawbacks, which are:

- a. In terms of scalability and memory management, it does not perform well.

This is because of its sub-optimal object-oriented design methodology: each node stands as its own object, thus creates a large number of dependencies, which are an obstacle when applied to large environments. Moreover on this issue, the authors in [30] and [31] report that a major deficiency of ns-2 is regards to memory handling as well as simulation run-time. This could be a difficulty for the field of computer networks that need large networks (hundreds to thousands of nodes)

- b. Lack of documentation.

As an open source simulator, ns-2 advances rapidly, yet does not always offer complete documentation (and it is often obsolete). This may complicate the learning of ns-2, since users need to familiar with the scripting language and modelling technique (TCL) which is somewhat difficult to understand and write. At the other extreme, it allows users (developers) to create their own scripting languages or extensions, resulting in different results for similar simulations.

- c. Graphical support.

In terms of graphical support, it does not have a proper GUI - the users have to directly face a text-based command interface. It is considered less user-friendly when compared to OPNET (which will be discussed later in the text).

- d. Lack of application model.

- (1) As described in [32], there are two different signal reception models in wireless transmission: SNR threshold and BER models. The SNR threshold model is used to accept a signal when its SNR value is above the SNR threshold, while the latter is based on a probabilistic decision over the frame length. In terms of this aspect, ns-2 only supports the SNR model, whereas OMNeT++ supports both models [33].
 - (2) In terms of layered networks, there are many interactions between protocol levels: cross layer systems are common. ns-2 was originally targeted at IP networks with limited degree of application behaviour, so there are constraints when applying it into layered networks (and more so when cross-layer aspects are considered).
- e. Its successor, ns-3, was still in the stages of development in 2009 [34]. This could be an alternative as recently much research work has been using this as their simulation model [35] [36] [37]. It still relies on C++, but oTcl scripts are no longer used here to avoid the problems when combining oTcl and C++ in ns-2. Also, it is possible to apply pure C++ in ns-3, while parts of the simulation optionally can be recognized using Python [34]. However, ns-3 is lacking in terms of backward compatibility with ns-2. Also, just a small number of models currently be present which can be applied off-the-shelf.

2.3.2 OPNET Modeler

As ns-2 has become established as one of the standard network simulation tools used among developers, another high-level discrete event network simulator named OPNET Modeler [38] [39] has also gained significant attention during recent years. OPNET (Optimized Network Engineering Tools) was originally a graduate networking project, which then became a commercial product. So it comes along with a bunch of predefined functions, protocols, devices and behaviours (approximately four-hundred), that

make it a powerful tool just from the start.

OPNET Modeler supports the use of modelling different sensor-specific hardware, such as physical-link transceivers and antennas. Several tools and editors are provided. It is a network level simulation tool which operating at a "packet-level". Originally built to simulate fixed networks, it is now available for wireless network hardware and protocols as well. It is also constructed from C and C++, and contains many network specific library functions.

Its hierarchical structure defines three domains which are:

a. Network/ topology level.

This level defines the overall scope of the system, for example a demonstration of a global world network. Some of the purposes of this are to identify and define the objects in the system, such as configurations, interconnections, locations, etc.

b. Node level.

This is where we define nodes such as routers, mobile devices, etc.

c. Process level.

This domain summarizes the behaviour of the modules, such as data traffic, packet processing algorithms, protocols, source code, etc.

Many research evaluations using OPNET [40] [41] [42] advise upon the basic correctness of this simulation tool. However, according to [43], its models always use a fixed topology. Also, we found its drawbacks to be in terms of a lack of object-oriented scalability, which makes it less flexible. It is not as popular as ns-2 and does not have high number of pre-defined protocols available.

Additionally, another major issue is that OPNET is not open-source software and is only available in commercial form. This consequently makes user-friendly

interfaces play an important role. A graphical interface (GUI) is provided and a lot of documentation comes along with it. Its graphical interface can help developers quickly simulate various models in graphical form as well as provide animation from the output: sometimes an important tool to aid the rapid understanding of such systems. Nevertheless, the cost of the software could discourage many developers.

2.3.3 OMNeT++

OMNeT++ [44] is the last simulation tool to be considered. Contrary to OPNET and ns-2, it is actually not a network simulator itself, but more of a general purpose discrete event-based simulation framework. However, because of its generic and adaptable architecture, it has been successfully used in other areas: network / hardware system architecture, queuing systems, or in the area of communication networks environment.

OMNeT++ is free for academic and non-profit use, and also has been utilized broadly in the global scientific community. Extension of framework models (MiXiM, Castalia, INET, etc) is provided for free. It has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into any applications. Many protocol models are also available in OMNeT++, such as: TCP/IP networks, peer-to-peer networks, LAN/MAN protocols, Ad-hoc, and WSN.

We make a comparison of this simulator by also considering earlier reports by [43] and others, summarised by the following points:

a. Protocol models.

OPNET has several prebuilt protocol models (i.e. Ethernet, TCP/IP, ATM), however it is expensive. We found that OMNeT++ is more accommodating (flexible) and mature [45] for protocol modification, besides with its other frameworks model available such as INET [17], MF [46], MiXiM [47], etc. Recent works in 2009 by [48] and [49] also demonstrate that OMNeT++ has gained

assurance being a simulator choice prior to implementing actual systems.

b. Hierarchical / Network Topology models.

ns-2 does not allow hierarchical models, which limits its use. OPNET allows hierarchical models, but only with a fixed topology. Since flexibility of different topologies and protocols is desired, OMNeT++ appears well suited to this. As mentioned in [50], although OPNET allows hierarchical models with arbitrarily deep nesting, OMNeT++ has more flexibility by permitting customized topologies to be used.

c. Debugging and GUI environment.

ns-2 debugging performance is considered relatively poor since it takes longer and uses lot of memory to simulate. Moreover, some analysis tools for ns-2 are not provided with the core simulator, i.e. they are available in separate patches (Visualization *nam*⁷, trace file analyser *Trace graph*⁸, and an analysis tool for wireless simulations *INSpect*⁹) that need to be integrated separately.

By contrast, OMNeT++, offers two different ways to trace files, using *Plove* and *Scalars* (both are provided as part of the software bundle). Furthermore, for the convenience, it can export debug data to other format like Microsoft Excel, MATLAB, etc.

OPNET possess a powerful command-line simulation debugger, but it does not boast a rich graphical runtime debugging environment, as OMNeT++ does. This convenient GUI-based tool makes the tracing and debugging much easier (again, we are considering the eventual end-users rather than the network researcher, as mentioned above when we started considering a simulation model).

d. Simulation comparison.

⁷<http://www.isi.edu/nsnam/nam/>

⁸<http://www.tracegraph.com/>

⁹<http://toilers.mines.edu/Public/Code/Nsinspect.html>

Two different simulations have been carried out for comparing OMNeT++ and ns-2. First, in [50], a simulation of Directed Diffusion under simple MAC and IEEE 802.11 MAC was performed. The parameters compared were delivery ratio, runtime, and memory usage. The concluding result showed an excellent performance indicated that OMNeT++ is much more scalable than ns-2 for execution time as well as memory overhead. Another simulation by [51], and similar report by [52] have shown that OMNeT++ outperforms ns-2 simulator regarding runtime performance and network sizes. A project by [53] showed that three different ns-2 implementations for the same protocol showed major differences in the results, leading to claims of inconsistency in the use of ns-2.

e. Language programming.

As mentioned earlier, in contrast to ns-2, OMNeT++ offers more flexible programming models. They are written in C++ language, supporting the full language syntax. Also, as parameters need to be changed during a simulation, certain information can be loaded dynamically¹⁰.

f. Open source and cost.

OPNET provides the source code for protocol models only, which is not preferable. Both OMNeT++ and ns-2 are fully open-sourced. This is the last but not least consideration for choosing our simulation tool. Furthermore, since OPNET is quite costly compared with the other two, we ruled out the further use of OPNET in this research project.

Based on the above explanation, then we choose OMNeT++ than ns-2 and OPNET as our simulation tool to be used in this thesis report.

¹⁰We will use this feature later when performing adaptive modulation switching, by dynamically change the parameters in the *ned* file.

Figure 2.1: OMNeT++ Model Structure

2.4 Modelling in OMNeT++

This sub-section will briefly explain OMNeT++ modelling. As mentioned above, OMNeT++ (Objective Modular Network Testbed in C++) is an object-oriented modular discrete event simulator mainly focusing on modelling both wired & wireless communication networks and protocol & queuing network.

Overall, it models and simulates any system where the discrete event approach is suitable, by conveniently mapping all the communications by exchanging messages. One of the fundamental ingredients of this infrastructure is its component architecture. Modules (components) are programmed using C++, then assembled into larger models using a high-level language (*ned*). Graphical, animating user interfaces are highly useful for demonstration and debugging purposes, and command-line user interfaces are best for batch execution. OMNeT++ supports both natively.

Components (modules) are programmed in C++ in a hierarchical style, and then assembled into larger components and models. The active modules are named *simple modules* (based on C++, using the simulation class library). It is possible to group simple modules into *compound modules* (refer to Fig. 2.1), and communicate with each other using so-called *messages* that contain data, via *gates*. Both input and output gates are linked up with a *connection* (also called *link*), to which we can assign *parameters*, such as bit error rate, data rate, propagation delay. Those parameters can take values like string, boolean, character, numeric, constants. Parameters can be assigned in either the *ned* files or the configuration file *omnetpp.ini*. Since it is based on C++, all the modules / sub-modules can pass parameters to each other and

to those that inherit their properties.

2.4.1 INET framework model

The INET framework is a popular open-source network simulation package built upon OMNeT++, thus it shares similar concepts. The advantage of using INET is that its framework contains well-known protocol implementations, such as TCP, UDP, SCTP, and several application models. More convenience in using INET comes from the fact that the packages are organized according to OSI layers, from the top to the bottom layer. Network devices such as routers, are represented by OMNeT++ using compound modules that are assembled from simple modules. It also contains models for both wired and wireless networking protocols models

We consider the INET framework as we are planning to implement systems employing existed protocol (i.e. TCP). Since INET has most of these built in, this is a good reason to use the INET framework. For more information regarding INET, please refer to [17].

2.5 OMNeT++ Simulation: Lost and Received Packets

2.5.1 Concept

In this section, we simulate a single wireless transmission link concept between two nodes (Node A and Node B) over the air, primarily in order to gain confidence in using OMNeT++ as a network simulator and to validate and explain its operation.

First, we derived analytical equations related to packet transmission over air, to be tested against the experiment to confirm that the OMNeT++ simulation is

working correctly.

We know that a total number of packets transmitted over the air should be equal to the number of packets arrived at the receiver, with any discrepancies between those nodes being the lost packets. From this concept, we start with:

$$N_P = L_P + R_P \quad (2.1)$$

with N_P is the total number of packets sent by Node A, L_P is the lost packets over the air, and R_P is the number of received packets at Node B.

2.5.2 Bit and Packet Error Rate

During (digital) transmission, there are possibilities that the transmitted packet (representative of bits) is being altered due to noise and interference. Bit Error Rate (BER) is a parameter used to state the number of bit errors over the total number of transmitted bits during a certain time period. It is commonly expressed as a unit of fraction / percentage to quantify a network performance. For example, for 1 bit error during 1000 bits transmission will result in a BER of 10^{-3} . BER may be affected by noise, interference, fading, attenuation, etc. It can be improved by several methods, such as by applying robust modulation schemes (discussed later in Chapter 6), or channel coding scheme (i.e. forward error correction codes).

Meanwhile, packet error rate (PER) shows the number of incorrectly received data packets divided by the total number of received packets. The relation between *BER* (Bit Error Rate) and *PER* (Packet Error Rate) as suggested by [54] is:

$$PER = 1 - (1 - BER)^N \quad (2.2)$$

with N being the uniform length of each packets in bits.

Figure 2.2: Simulation topology.

Lost packet L_P indicates the packet error rate percentage of total number of packets sent:

$$L_P = PER * N_P \quad (2.3)$$

Then from (2.1) and (2.2) we can relate the total number of packets sent and lost as:

$$N_P = \frac{R_P}{(1 - PER)} \quad (2.4)$$

and

$$L_P = \frac{R_P}{(1 - PER)} - R_P. \quad (2.5)$$

Later we test and prove the equation derived by theory and the results from the experiment of those equations as described below.

2.5.3 Experimental scenario

We construct the simulation by establishing two nodes, Node A and Node B which communicate through wireless link. Node B will stop the simulation once it receives n -correct packets (equal to the value of R_P) regardless of the number of packets sent/resent by Node A. The environment (depicted in Fig. 2.2) was simulated under the variable BER circumstances, verifying the channel statistics as the simulation

Figure 2.3: Lost packets vs. BER, for $N = 1024$ & 2048 bits.

proceeded.

The simulation was carried out with the parameters given in Table 2.1. We varied the BER of the channel as the simulation progresses, with predefined packet sizes and total correctly-received packets. The results of this simulation, confirming the theory of (2.4) & (2.5), are shown in Figures. 2.3 and 2.4

2.5.4 Results and discussion

We discuss the comparison of lost and sent packets, both are related to the BER condition for two different packet length. In general, the simulation was run and the results are plotted against the theoretical value.

Table 2.1: Simulation-1 parameters

Attribute	Value
Rp (packets)	125
BER	10^{-6} , 10^{-5} , 10^{-4} , 10^{-3}
N (bits)	1024, 2048

Figure 2.4: Sent packets vs. BER, for $N = 1024$ & 2048 bits.

As (2.2) suggests, with a smaller packet length, i.e. $N = 1024$ bits, the packet error percentage will be lower compared to the lengthier packet. Thus small packet error rate will result in the loss of fewer packets, as suggested in (2.3). This can be observed clearly at Fig. 2.3, particularly at constant BER value $= 10^{-3}$. And as expected, packets with bigger size (i.e. $N = 2048$ bits) clearly showed stronger tendency to become lost.

More details on the BER observation, when progressing from good to bad, the results show a substantial increase in the number of lost packets. For packet with bigger size, most of the packets were lost (811 packets compared to 213 packets for smaller size packet) for the worst BER value. On the contrary, for good bit error rate condition, the number of lost packet was very small (approximately zero). These observations were as anticipated.

From the point of view of sent packets, as shown in Fig. 2.4, the encouraging result was achieved. For bad BER, more packets were delivered from Node A (from 338 to 936 packets) to convey the differences amount of packet size. When it comes

to a decent error rate value, the number of sent packets was equivalent to 125 (the R_P value we set as given in Table 2.1), indicating that no packets were missing during the transmission.

From this section, it can be concluded that the simulation results have showed that the OMNeT++ simulation results are consistent, since the discrepancy between theory and experiment is very minor.

2.6 Summary

To sum up this chapter, we have evaluated several network simulation tools as summarized in Table 2.2 prior to chose OMNet++ as our tool for subsequent investigation. An evaluation model scenario using OMNeT++ is performed, i.e. a simulation of lost and received packets with respect to bit and packet error rates. An encouraging result and conclusion was obtained. Theoretical analysis was used against our simulation model, and it could be perceived that the discrepancies between experiment and theory are very negligible, and this builds assurance in the consistency of OMNeT++ as a network simulation tool to be used for this research.

Table 2.2: Network simulation tools comparison

	<i>ns2</i>	<i>OPNET</i>	<i>OMNeT++</i>
General use	Primarily used in research & teaching	Originally a networking tool. Since Dec. 18 2012 acquisitioned by RiverbedTechnology – become an application and network performance management market	Simulation of communication networks, also gain popularity in scientific and industrial community
Language	C++	Proto-C	C++
Documentation	Not so complete / fragmented (short tutorial available)	Good, but need to purchase	Organized (short tutorial available)
Open source	Yes	No	Yes
Performance	Good	Good	Good
Popularity	Very popular	Un-popular	Popular
Scalability	Low	Fair	High
Visualization/GUI	Fair	Good	Good
Signal reception model	SNR-based	BER-based	SNR- and BER-based
Successor / Extension	ns3 (not backward compatible)	Not available	Packet/extension available (INET, MiXiM, etc..)
OSI layer	Available	Available	Available
Structure	Flat / tight coupled	Hierarchical 4-layer	Hierarchical (easier implementation)
Self-develop protocol	Yes	Yes	Yes
Application / Simulation model	Decent	More features than ns-2	Good variety
Protocol model	Rich set	Vast number	Reasonable many, but lags behind ns2

Chapter 3

Segmentation and Reassembly with Error Control mechanism

3.1 Overview

In this chapter, we extend our simulation model to be more realistic as we consider a concept of segmentation and reassembly (SAR) in addition to error control mechanism. We use similar topology as used in the previous chapter with adding a parameter called block size, and then vary the scenario accordingly. We also consider bit transmission efficiency as another way to evaluate our model. Discussions are given following the modelling.

3.2 Segmentation and Reassembly

Segmentation and Reassembly (SAR) is one of the concepts used to fragment and reaccumulate packets, allowing them to be safely transmitted across error-prone transmission links. The process is straightforward, as the incoming packets to be transported are chopped up into smaller segments (along with the desired size), then augmented

Figure 3.1: Concept of segmented data and block size

with a header, and perhaps some error code/check information. At the other end, these segments are then rebuilt back together forming the original transmitted packet. Manageable sized packets are thus usually segmented in the source node and reassembled at the destination node.

As the realization of the concept above, we introduce a block size parameter, w , defined as the length of segments that we transmit over the air. The data to be transmitted is grouped to fix the size of w . A one-byte sequence number (SN) is added at the beginning as the header of the block size. Consequently, SN will increase following the size of the data. The concept can be seen clearly in Fig. 3.1.

3.3 Error control mechanism: Retries and ReACKs

To be more realistic in modelling, we added an error control mechanism [55] (just as TCP does). We limited our implementation in terms of number of re-tries and re-ACKs. For every segment sent and received, both Node A and Node B will set a timeout counter, which is exclusive for each segment. Once the timeout expires, each node will perform its error control mechanism, i.e. Node A will retry the segment if it does not receive an appropriate acknowledgement for the corresponding segments,

Figure 3.2: Error control mechanism: re-ACK scenario once the timeout expires

or as shown in Fig. 3.2, Node B will re-send the ACK since the timer is up (meaning its previous ACK was lost).

Further in the implementation code, once this timer value expires the receiver will assume that the (previous) ACK has been lost. So it will re-send the same ACK number packet with newly ACK-timer value. Note that in the model, the ACK-timer value is adjusted so it will not trigger a resending packet routine from the transmitter. These mechanisms was designed to ensure the segments were sent and received correctly, thus increasing the realism of the model and simulation¹.

We also set the number of per segment retries and reACKs parameters to 3 and

¹Though Radio Link Protocol (RLP) can make use of either ACK- or NAK-based error control mechanisms, this thesis does not use the NAK protocol for two main reasons. First is to minimize the computation complexity in the receiver, i.e. the frames successfully received are guaranteed correct in-order to ease the progress of reassembling. Second, the NAK-based protocol implicitly indicates the need of a significant output buffer to facilitate the correct reassembly and assumed-acknowledged out-of-order frames. By directly implementing the reassembly process as this thesis does, and then immediately outputting known good packets, this issue can be disregard.

NAK certainly has advantages, and it is probably one implementation choice that might be made in practice. However the focus of this thesis is on the effect of *other* methods of improving goodput based on packet handling. We implemented the simplest possible working baseline system for testing. In fact, the improvements (or degradation) due to different methods of packet handling are independent of NAK/ACK method.

5, respectively. If both retries and reACKs exceed this value, then the segments are considered lost.

3.4 Bits transmission efficiency

In addition to the mechanism above, we are interested to find out how many segments are received correctly over the total transmitted segments from the sender. It is possible that during the transmission some bits of the segments are experiencing an obstruction. Thus we created another parameter, bit efficiency:

$$\eta = \frac{\text{Received bits}}{\text{Total sent bits}} \quad (3.1)$$

This can be used to determine the maximum bits per second we can transmit over the air and to work out how many seconds we need in order to transmit a certain amount of data.

3.5 Experimental scenario: Realistic wireless transmission with error control

The modelling started with a simple concept of segmentation and reassembly (SAR). The same topology was used as previous chapter, but with a slightly different scenario as follows:

- a. Node A states the number of segments according to the length of the data, by breaking it into the appropriate block size w (i.e. segmentation process) then adding one byte sequence number prior to beginning to transmit to Node B.
- b. Node B will be receiving all the segments. Once all the segments were correctly arrived, they will be re-assembled and full data will be generated.

Figure 3.3: Nodes A & B flowchart

- c. In addition to SAR, at this time Nodes A and B have an error control mechanism. Node A must re-deliver the lost segments (retries) during the transmission, while Node B had to re-deliver the acknowledgement (reACKs) when it received the same segment from Node A (this means the previous acknowledgement is lost). The flowchart of this scenario is depicted in Fig. 3.3.
- d. The maximum number of retries and re-ACK were set to 3 and 5 respectively (which can be modified easily).

The simulation was conducted with the conditions stated in Table 3.1, while the results were plotted in Fig. 3.4 and Fig. 3.5 in terms of BER and PER for different data sets, while the results for number of retries & reACKs along with block size parameter were plotted in Fig. 3.6 and Fig. 3.7.

Table 3.1: Simulation-2 parameters

Attribute	Value
Block size, w (bytes)	49, 196
Data length (bytes)	400, 1600, 4000, 8000, 12000, 20000
Retries and reACKs	3, 5
BER	10^{-5} , 10^{-4} , 10^{-3}

Figure 3.4: PER vs. BER of various data sets, with $w = 49$.

Figure 3.5: PER vs. BER of various data sets, with $w = 196$.

3.6 Results and discussions

Firstly, in the relation between BER, PER and block size, the results in Fig. 3.4 and Fig. 3.5 showed that, as expected, a channel with bad condition caused more percentage of data with errors will be lost in transmission – and the same tendency happens when we varied the data size. This, again, confirming (2.2), thus showing a large percentage of packet error rate.

The value itself increases gradually as we changed the block size from 49 to 196, as can be seen from both figures. Based on this observation, a cautious choice of block size has been advised to help maximize system efficiency [56]. A small block size implies a greater overhead in terms of segment header and ACK traffic.

In terms of retries and reACKs, Fig. 3.6 and Fig. 3.7 showed that more segments apparently were re-sent under bad BER conditions. It is interesting to notice that for different block size, the plots depicted similar values – this can lead to the assumption that increasing block size does not automatically lessen the number of retries and reACKs.

Lastly, we altered the parameters in our experiments to determine the association between number of retransmission (both retries & reACKs) and:

1. PER percentage of received segments,
2. total bits sent over the air, to analyze the bit transmission efficiency.

We modified the simulation structure and operating parameters, changing the value of retries/reACK from 3 to 5 to identify the correlation. We narrowed our discussion to relatively poor BER value of 10^{-3} , with the results are shown in Tables 3.2 and 3.3 respectively.

Table 3.2: Comparison of PER & total bits sent on 3 retries, BER= 10^{-3} and $w=49$.

Data Length (bytes)	3 Retries				
	PER (%)	Bits received	Node A retries	Node B reACKs	Efficiency (%)
12000	1.22	94424	123	15	98.36
50000	1.18	393696	499	68	98.42
100000	1.37	785824	995	129	98.23

Table 3.3: Comparison of PER & total bits sent on 5 retries, BER= 10^{-3} and $w=49$.

Data Length (bytes)	5 Retries				
	PER (%)	Bits received	Node A retries	Node B reACKs	Efficiency (%)
12000	0.0	95600	128	15	99.58
50000	0.1	398008	516	68	99.50
100000	0.1	796016	1037	129	99.50

Figure 3.6: Total Retries and ReACKs vs. BER, with $w = 49$.

Figure 3.7: Total Retries and ReACKs vs. BER, with $w = 196$.

Once the maximum number of retries was increased from 3 to 5, we noticed that packet error rate percentage reduced, which means that all the segments had a greater probability of reaching at Node B. We also noticed that total bits sent over the air increased (which also can be seen from the increasing number of total retransmission (retries and reACKs)). This makes sense as more retransmissions happened, more segments are sent over the air, and therefore the number of bits of air traffic increased.

Table 3.4: Comparison of PER & total bits sent on 3 retries, $\text{BER}=10^{-3}$ and $w=196$.

Data Length (bytes)	3 Retries				
	PER (%)	Bits received	Node A retries	Node B reACKs	Efficiency (%)
12000	41.94	55232	123	3	57.53
50000	39.45	241632	490	9	64.41
100000	38.75	487960	976	22	61.00

Table 3.5: Comparison of PER & total bits sent on 5 retries, $\text{BER}=10^{-3}$ and $w=196$.

Data Length (bytes)	5 Retries				
	PER (%)	Bits received	Node A retries	Node B reACKs	Efficiency (%)
12000	27.42	69344	180	3	72.73
50000	22.27	309048	671	12	77.26
100000	23.48	608688	1324	26	76.09

The simulation result showed similar tendencies when we increased the block size to 196 bytes (shown in Table 3.4 and Table 3.5 for retries/reACKs equals to 3 and 5 respectively).

The bit transmission efficiency was also analyzed by comparing the total received bits over the total data length as suggested in (3.1). We found that for the tested parameter ranges, efficiency decreased when the block size increased from 49 to 196, for both retries value of 3 (see Table 3.2 and Table 3.4) and 5 (see Table 3.3 and Table 3.5). This was due to the straight correlation that when part of any segment – determined by block size – is lost, so are all of the bits contained within it. So bigger block size more likely make the segment being lost is higher. In reality, even relatively weak error check coding (ECC) could improve this, but at the expense of greater node CPU speed requirements, greater RAM size, and possibly increased transport latency.

Nevertheless, we discovered something interesting about the number of sent bits when we extended the block size from 49 to 196: the amount of lost bits nearly reached 42%; an extremely large overhead. From this we noted that there was a trade-off to be regarded in terms of block size and bit transmission efficiency within any implemented system.

3.7 Summary

To sum up this chapter, we have successfully simulated a simple wireless transmission concept by developing a segmentation and reassembly simulation model incorporating error control mechanism (i.e. retries and reacknowledgement). The simulation was carried out for varied data size in the condition of channel's error rate. Moreover, bit transmission efficiency also being analyzed in relation to the block size parameter and the number of retries and reACKs.

All the results are showing very minor in terms of the differences between theory and experiment – thus gained more confidence in the OMNeT++ simulation model. Furthermore, the model developed in this and previous chapter will be used as the basis of further our OMNeT++ simulation.

Chapter 4

TCP-Based Transmission Model

4.1 Overview

This chapter will elaborate on the well-known TCP protocol and the motivation behind using it in our model. Several previous improvements and progressions of this protocols are also discussed prior to build our TCP-based point-to-point transmission model. To give a more realistic environment, we combine the concept of SAR and error control mechanisms as explained in the previous chapter. We introduce two important parameters which are segment size and maximum number of retransmissions as a means to facilitate those concepts. The results of our simulation are discussed systematically before concluding the chapter.

4.2 Classical wired vs. wireless protocol

It is generally understood that protocol implementation for wireless networks is still an open research issue. This may stem from several differences between classical and wireless networks. In classical networks, packets are considered as independent byte streams; and the network is represented as having packets travelling through multi-

hops nodes. This is precisely what the classic transport protocol (TCP) provides, transporting packets from source to destination node (end-to-end) by implementing some mechanisms, for example segmentation and reassembly (SAR), flow and error control.

However, in wireless networks, conditions are often relatively different: instead of an individual (end-to-end) perspectives, each node pairs interact (dependently) with a hostile transmission environment, such as unpredictable wireless channel (fading, interference), node(s) mobility, vulnerable media access, as well as mechanisms like reflection, diffraction, and scattering that affecting the signal propagation through the channel [5] [57]. Furthermore, higher error rates are expected, thus the previous assumptions of congestion become problematic since the packet-losses are mainly caused by high error rate transmission, not only congestion as is typically the case for wired networks.

Standard transport protocols are typically used for improving network congestion, minimizing packet-loss, and assuring end-to-end reliability. They generally work well in wired transmission systems.

4.3 Standard transport protocol

The TCP/IP protocol suite¹ specifies two protocols for the transport layer, which are the User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). The relationships between those protocols is shown in Figure 4.1. They lie between the application layer and the network layer, serve as an intermediary between application programs and network operations.

The transport layer usually has several duties. First, it is to make a connection mechanism for process-to-process communication. Viewed from above layer, the

¹Details on TCP/IP protocol suite can be read in [58].

Figure 4.1: TCP/IP protocol suite: TCP and UDP

sender has a responsibility to make sure that streams of data from the network layer are sent to the transport layer. Once all units have been sent, the transport layer will close its connection. Secondly, another role of transport protocols in general is to provide control mechanisms for data flow and error control.

For UDP, it performs the task at a very minimum level: no error control mechanism, no flow control, and thus no window mechanism is provided. The receiver may overflow with incoming packets, while the sender will not know if there is a retransmitted or lost packet. The receiver silently drops a packet if it detects any error.

Given the above traits, UDP is called a connectionless and unreliable transport protocol, which mean that it has the usual transport layer responsibility (control mechanism, process-to-process communication) at very minimal levels. Every single packet is considered as an independent packet (datagram), and there is no relationship between the present and incoming datagram. It does not convey the concepts of a sequence number, connection establishment, and connection termination. Each user datagram is considered to have potentially travelled on a different path.

Although, being unreliable and loosely controlled, it does not appear useful, in fact there are several motivations for using this protocol:

1. Minimum effort and overhead: means that UDP is a very simple protocol which initiating less interaction transmissions between the sender and receiver (compared to TCP). This behaviour is accurately showed in a up to date work by [59] by showing that performance of the TCP protocol rapidly decreases with packet loss increment while the UDP protocol keeps performing effectively in difficult conditions. The explanation is quite apparent that when the well-known TCP handshake is performed, several packets must be exchanged without error even before data transmission occurs. In high error rate environments this becomes troublesome.
2. It is considered fit when error correction and checking is not essential. For example, for time-sensitive applications, use of UDP is sensible where dropping a packet is preferable to waiting, which may not be an option in a real-time system [60].

Other recent research shows that the performance of UDP protocol is lower than TCP in many other situations. For example, [61] showed that the PDF (Packet Delivery Fraction) percentage of three MANET protocols (AODV, DSR, and DSDV) generated in constant-bit-rate are still lower if using the UDP protocol. From real MANET experiments and simulations of TCP and UDP, [62] showed that TCP throughput performance still outperforms UDP in many real scenarios.

4.3.1 TCP

TCP² (refer to Fig. 4.1) acts as the liaison of lower network layer operations and higher application layer programs. It has a duty to provide a connection mechanism for the application program: when the sender is sending streams of data, initiating connection, splitting the data, adding a sequence number until the data has been

²Details on TCP protocol can be read in [55].

received correctly at the receiver (where it is to be checked prior to closing the connection) - TCP handles the entire process. This is the reason why it is called a connection-oriented and reliable transport protocol, in contrast to UDP.

In terms of flow control, TCP can define the quantity of data that a sender can send before getting an ACK from the receiver. For example, TCP could send 1 byte of data only and wait until an ACK is received prior to sending the next byte. And on the other hand, it could send all of the data without concerning about the ACK. Thus if the data somehow is corrupted, duplicated, or lost, the sender will not know. It will make the communication more efficient over an error-prone channel since this protocol will be able to control the flow of data.

TCP also offers reliability using error control and error detection mechanisms. For error control, its purpose is to detect wrong segments such lost, corrupted, or out-of-order segments, while for error detection, it uses ACK to verify if the transmitted segments have reached the destination safely and uncorrupted. Since TCP does not use NAK (Negative ACK), a segment is considered lost or corrupted if it is not ACKed before timeout.

4.4 TCP over wireless links

TCP is the prevalent protocol for data transport over the classical wired Internet. Unlike UDP, its purpose is to make a reliable end-to-end data transmission link and to control network congestion over potentially unreliable networks. As known, the present TCP protocol generally interprets all losses as being caused by packet congestion. Whenever losses happen it reacts to this as though losses were due to congestion, thus it decreases the network transmission rate or throughput. In general, this issue can mitigated by retransmitting lost packets and reducing transmission rate (although a number of other possible solutions exist).

4.4.1 Drawbacks

In the literature, several researchers have put emphasis on discussing the drawbacks of using TCP in wireless environments, for example [63] [64] [65]. [66] claimed that TCP is not the best option for obtaining high throughput performance from wireless links, especially in high error rate situations. This is mainly because TCP assumes that packet-loss is due to congestion, and responds by reducing its transmission rate. Whereas, in fact, packet-loss is due to bit errors. Moreover, [67] and [68] stated that, in terms of overhead, TCP might not be justified for it employs loads of bytes (in contrast with a small data packet requirement in sensor networks).

In relation with TCP mechanisms:

- i. The well-known TCP connection establishment and termination mechanisms (Fig. 4.2) would make a network more likely to become congested.
- ii. TCP is responsible for providing network reliability by performing acknowledgement (ACK) and retransmission mechanisms. However, in some implementations, this could mean more ‘extra’ packets are travelling across the networks. This is at the cost of additional power and bandwidth.
- iii. Flow and congestion control mechanisms in TCP might be tough for distant nodes to handle effectively. Again, this will cause imbalanced bandwidth allocation.

4.4.2 Improvements

Apart from the explanation above, research has been carried out to improve TCP performance over wireless networks, and many alternative solutions have been proposed:

1. Indirect TCP (I-TCP) [69].

This idea was performed by splitting the system into two connections, both

Figure 4.2: TCP connection and establishment.

wireless and wired medium. The TCP sender is made unaware of the loss of a wireless link; consequently it would not invoke congestion control mechanisms. However, as reported in [70], this approach would breach the TCP semantics, as acknowledgement most likely arrives at the sender before packets arrive at the destination. To prevent this situation, a delay in acknowledgement is required, thus implying reducing in throughput.

2. TCP Feedback (TCP-F) [71].

The second proposed scheme was using a feedback approach. It introduces an intermediate mobile host, beside the sender and receiver hosts. When a route disruption happens in wireless links, an intermediate mobile host will send a *route failure notification* (RFN) back to the sender. The sender will switch from *active* to *snooze* state, at which point it completely stop sending packets and freezes all its timers and internal values. It remains waiting until the link is restored by receiving a *route re-establishment notification* (RRN) from the intermediate host, then the process will resume sending the packet.

3. TCP with explicit link failure notification (TCP-ELFN) [72].

Another scheme has been developed, similar to TCP-F. Instead of RFN, the sender receives an *explicit link failure notification* (ELFN) before it goes into *stand-by* state (*snooze* in TCP-F). The difference is, rather than the intermediate node, the sender node periodically sends a small packet to the network to see if a new route has been found. Both TCP-F and TCP-ELFN mechanisms indeed will enable the sender to resume fast transmission. However, according to [73], they did not consider the effect of congestion, bit error rate issues, and obsolete packets.

4. Adhoc TCP (ATCP).

A better feedback scheme than TCP-F and TCP-ELFN was developed by [74]. In this, the sender separates the network state information through two notification messages:

- i. When receiving a *destination unreachable* message, the sender will go into *persist* state, by freezing and not invoking congestion control mechanisms until a new route is found.
- ii. On getting an *explicit congestion notification* (ECN) message, TCP senders go into *congested* state, then invoking congestion control mechanism immediately. On the other hand, if packet losses still happen and the ECN flag is not set, there are two possibilities for this: high bit-error-rate or receiving duplicate ACK (out-of-order delivery). Simply the sender will re-send the packet (*retransmit* state).

As can be seen, this approach is considered more comprehensive, since it utilizes more possible wireless flaws: congestion, bit error, and out-of-order delivery.

5. Lower-layer enhancement, as reviewed by [73].

This scheme is based on the idea that lower layers such as routing and MAC

layers need to be adapted in light of the knowledge of TCP's congestion control algorithm:

- a. In the routing-layer, a work by Anantharaman et al. [75] proposed 3 mechanisms to minimize the probability of route failures, to predict route failures in advance, and to minimize the latency in conveying route failure information to source, respectively.
 - b. In the link-layer, a report by [63] suggested two methods to improve TCP efficiency, which are Link-RED (Random Early Detection) algorithm and Link-layer pacing schemes. Basically, the link-RED algorithm has a purpose of maintaining the optimum TCP congestion window size, while the latter is to keep alleviating the medium contention.
 - c. Another link-layer scheme named Neighbourhood Random Early Detection (NRED) [76] introduced three algorithms which in essence, the main idea is to incorporate one particular nodes neighbourhood for cooperation with, instead of employing the mechanism single-handedly. It is reported that NRED can improve TCP fairness to some extent in ad-hoc networks, though the network throughput is reduced.
 - d. Balakrishnan et al. tried to 'smooth' the link by using appropriate channel coding and ARQ so that the link is perceived as a constant channel, but with lower capacity [66]. Yet the problem still exists that the TCP sender may not be fully protected from wireless link losses. And again, TCP congestion mechanisms will be invoked, causing loss of throughput and retransmission.
6. Adhoc Transport Protocol (ATP).
- A work by Sundaresan et al. in 2005 suggested a new mechanism of transport protocols (similar to TCP) named ATP [65]. The results claimed that the protocol accomplishes much better performance in terms of delay, throughput,

and fairness, than TCP variants since it uses a new mechanisms. In their work, we found compatibility issues to be noted, however. ATP assumes that the wireless network should be stand-alone and not to be integrated with many other wireless networks, and this type of restriction is not favoured in existing wireless networks as mentioned before.

4.4.3 Remarks

We have seen and discussed the possible drawbacks of using TCP as well as encouraging improvements that have been made in terms of enhancing TCP for wireless environments. To date, none of them are guaranteed to work well in all possible wireless scenarios (obviously). Besides, in the future, systems naturally tend to scale in size, becoming more complex and larger over time. It is not clear yet what people will use as a reliable and absolute transport protocol for future wireless systems. We are considering re-visiting TCP as our wireless transport protocol since it is mature, ubiquitous, and proven-reliable in wired environment, as well as the fact that there are still many open opportunities and challenges for research in this area³.

4.5 TCP-Based Point-to-point Transmission Model with SAR and Error Control

Moving forward to implement the concept, we build a wireless link model which adopts the TCP protocol. This is the basic model employed in this section, combined with segmentation and reassembly (SAR) concepts for TCP packets, allied with an ARQ-based sliding window error control mechanism to reduce packet error rates (PER) visible to the TCP layer. The model is simulated using the INET framework [17],

³As performance degradation mostly lies in the physical layer, MAC layer, and network layer, cross-layer design approaches are worth investigating [77], although with some caution [78]. More of this discussion can be found in Chapter 6.

Figure 4.3: Layer and data handling structure of single-link TCP-based point-to-point model

an open-source network simulation package for the OMNeT++ environment. As it contains models for both wired and wireless networking protocols, including TCP/IP and 802.11 models, we adopt this for the baseline simulation throughout this thesis report.

For the basic structure, we employed a standard layer-oriented data handling arrangement in the system, as shown in Fig. 4.3. From a users' perspective, we define two wired nodes with TCP connectivity (Nodes A and B) which are to communicate over a wireless link. We then 'insert' two wireless transmission-nodes (X and Y) from our previous model in Chapter 3⁴. Both Nodes X and Y connect to Nodes A and B by wired links respectively, and communicate between themselves wirelessly as shown in Fig. 4.4.

The model complies with all of the well-known TCP/IP procedures, i.e. connec-

⁴Published in [79].

Figure 4.4: TCP-based point-to-point transmission model.

Figure 4.5: Encapsulation and Decapsulation concept.

tion establishment and termination, flow and error control mechanisms, and also the concept of encapsulation and de-capsulation from the layers above and below (Fig. 4.5), while Fig. 4.6 show its implementation in this report.

4.5.1 Segmentation and Reassembly

Node A transmits TCP/IP data to node B through intermediate nodes X and Y, which are responsible for ‘hiding’ the potentially high BER wireless link X–Y from Nodes A and B. Within the simulation, a large data block to be transferred to Node B is first split into a sequence of TCP frame or payloads at Node A. If this sequence is then transmitted and received by Node B without error, it will be reassembled into the received data block.

Figure 4.6: Decapsulation and Segmentation process implementation.

Figure 4.7: Concept of sequence number and segment size, and relationship to TCP/IP frame and original file.

In detail, the TCP payloads first reach Node X where they are split into smaller segments (determined by the segment size parameter, SS). A one-byte sequence number ($SequenceNumber$) is appended to each segment, forming a ready-to-send frame as parameter block size, w (similar with parameter used in Chapter 3, before it is transmitted over air from Node X, and received by Node Y. This arrangement is shown diagrammatically in Fig. 4.7.

4.5.2 Error Control

We also add another error control mechanism, similar to the stop-and-wait ARQ protocol (just as TCP does) which is limited in terms of the number of retries only (note: we included re-ACKs in our previous model as described in Chapter 3). These retries are used in conjunction with a timeout: for each segment sent, the sender (Node X) will not send any further data until it receives an appropriate ACK signal for the previous segment. For any case where a segment is missing (lost or corrupted), the sender will keep transmitting the same segment until it reaches the allowed retries value, r_{MAX} . (This will be explained later in the chapter while the concept can be seen in Fig. 4.8). Once this value is achieved, then all of the segments of the current payload will be discarded; then the transmission will start over for the same payload again.

This was designed to ensure that segments are sent and received correctly, since

Figure 4.8: Concept of maximum retry mechanism, r_{MAX} .

such techniques can be found in implemented systems. While more complexity and intelligence is added in later sections of this thesis, the underlying transport mechanisms remain predominantly as described above.

4.5.3 Evaluation methods

Within our evaluation, throughput and goodput are used for measurements to assess the effects of adjusting and changing parameters (several adjustment types are possible in this system, to enable different trade-off dimensions). Throughput TP is defined as the total number of bytes in a frame transmitted over the air and collected at the receiver per-transmission; these include retransmission frames and headers (refer to Fig. 4.9). Let $n + m$ be the total transmission over time duration t_T , calculated in Mbps or Kbps or bps, then throughput:

$$TP = \frac{(n + m)(N + 1)}{t_T} \quad (4.1)$$

By contrast, goodput is a measure of the amount of useful data received per

Figure 4.9: n separate segment transmissions, each containing N bytes of data, are made. Each segment includes a one byte sequence number. Due to errors, some segments need to be re-transmitted, maybe more than once. So m segment re-transmissions will be necessary to successfully convey all data.

unit time:

$$GP = \frac{nN}{t_T} \quad (4.2)$$

This is the application level throughput experienced by the user i.e. it only considers the successfully received bytes of the payload at the receiver – excluding protocol overhead and retransmitted frames. Furthermore, if any one part (segment) of a TCP/IP frame is missing after all retries have been exhausted, then the entire TCP/IP frame is discarded – even those other segments that have been received correctly – and none of the good data (from the partial TCP/IP frame) will be counted as goodput.

Table 4.1: Single-link simulation parameters

Attribute	Value
Transmitted file size	30 MB
Max. TCP-payload retransmission	12x
TCP flavour	TCP Reno
Channel data rate	256 kbps
BER	10^{-4} , 10^{-5} , 10^{-6}

4.5.4 Discussions

Modelling

This section will show the simulation model set up to assist in deciding the most effective segment size (SS) and maximum retransmission (r_{MAX}) to use for a particular data bandwidth, error rate and simulation conditions (which, in general, match previous research findings [15]).

The evaluation begins by varying parameter SS under several different BER scenarios, according to the parameters listed in Table 4.1. We used TCP Reno flavour as the basis of our model, with a total file size of 30 MBytes to be transmitted. To regulate the system, we limited the maximum TCP-payload transmissions to $12\times$ and used a channel datarate of 256 kbps as the default value. These chosen parameters can be easily modified, but are reasonable first estimates for the system [15].

The simulation was run for segment sizes of 64, 128, 192, 256, 384, and 512 bytes. We then plotted the achieved goodput for those segment sizes based on three bit error rates conditions: 10^{-4} , 10^{-5} and 10^{-6} (again, matching other published scenarios [15]). The maximum number of segment retransmissions is limited to 0, 2, 3, 4 and 5 times before an error is flagged.

The results are plotted in Fig. 4.10, where it can be seen that significant improvement of goodput is possible through use of a retransmission scenario under most of the tested BER conditions (i.e. performance is better when $r_{MAX} > 1$).

(a) Goodput vs. r_{MAX} , $SS = 64$. (b) Goodput vs. r_{MAX} , $SS = 128$. (c) Goodput vs. r_{MAX} , $SS = 192$. (d) Goodput vs. r_{MAX} , $SS = 256$. (e) Goodput vs. r_{MAX} , $SS = 384$. (f) Goodput vs. r_{MAX} , $SS = 512$.

Figure 4.10: Goodput comparisons for different segment size and maximum number of retransmission.

BER analysis

For all sub-figures, it can be seen that for $\text{BER} = 10^{-6}$, goodput results have almost a similar value, particularly when the retransmission is forced to be greater than $1 \times$. The explanation is apparent that given more chance to transmit the packet, there is a higher probability that the packet will arrive safely at its destination. Also, since this was mainly in a relatively good BER environment, it is most likely the goodput result of r_{MAX} between 2 to 5 will be similar. This tendency happens for all of the segment sizes value: in general we achieve diminishing returns for increasing r_{MAX} . $\text{BER} = 10^{-5}$ appears similar the $\text{BER} = 10^{-6}$ case. The difference is just the goodput values which are a bit lower.

These results help in verifying two things: first, that the retransmission scenario for varied r_{MAX} value is valid as the larger values yields higher goodput – particularly in a good link condition. And second, as the consequence of the former, increasing the r_{MAX} value should be taken with full consideration, since it will increase the overall transmission time; since it is possible that packets will need to be re-transmitted more times, in fact, it can be seen that often a choice of lower r_{MAX} provides the same goodput result. Again, this is applicable for the better BER links only. We note this as one of the aspects that should be considered when designing a real system to operate under such conditions.

Focusing our discussion into the $\text{BER} = 10^{-4}$ case, we found more interesting results. In general, the goodput value varies in a logical way. For $SS = 64, 128, 192$, and 256 bytes, high goodput is achieved most for $r_{MAX} = 4$ and 5. The no-retry scenario simply makes the system useless as no goodput is achieved at all in this low BER environment (note: results are not plotted for all SS scenarios). $r_{MAX} = 2$ and 3 is not advised, except maybe for a system with $SS = 64$.

For $SS = 384$ and 512, the goodput is encouraging as expected, and that as r_{MAX} gradually increases, so does the goodput – even though its absolute value is not

as high as that for segment size = 64, 128, 192, and 256 bytes.

Maximum retransmission and Segment Size analysis

For the effective r_{MAX} , after analysing the results in more detail, we concluded that the best minimum value of maximum retransmission is at $r_{MAX} = 4$. We argue that for the following basis:

1. For almost all SS values in the simulation, $r_{MAX} < 4$ yielded reduced goodput (we note this to be an improvement on previously published work that did not specifically include the effects of the TCP protocol [15]).
2. Assuming for $BER = 10^{-4}$ as the reasonably bad BER condition, we notice that r_{MAX} values of 4 and 5 give similar results (except for $SS = 384$ and 512 bytes as the differences is negligibly insignificant). A value of 5 would be a waste since having bigger r_{MAX} will only increase the possibility of yielding a longer transmission time. Thus using $r_{MAX} = 5$ would be ineffective. It extends worst-case transmission time yet provides little discernible benefit. It may even cause the transmission time to extend beyond the application-set limit for the TCP packet (in which case the higher layer would declare a time-out, and cause the entire TCP packet to be retransmitted – this is obviously a disastrous scenario for real system performance over poor channels).

Taking in account that there is an omission for $SS = 384$ and 512 bytes. For these, we argue that, to be compared with other SS values for the same $BER = 10^{-4}$, we found the lowest goodput value (3447 bps) occurred with a segment size of 384 bytes, which argues against this value being used. For $SS = 512$ bytes respectively, it is very apparent that its goodput is lower than any other moderate segment sizes (64, 128, 192, and 256 bytes).

Figure 4.11: Average *goodput* vs. *SS*

On the other side, for $SS = 128$ bytes, again at $BER = 10^{-4}$, maximum goodput is achieved (8702 bps). Though it is very similar to the value for $SS = 256$ bytes (i.e. 8662 bps), we consider the difference in overhead between these SS to be negligible – thus smaller segment size (in this case $SS = 128$) bytes is generally preferable.

Furthermore, it is interesting to note that for worse BER, the system is still able to achieve reasonable goodput, but at the cost of $r_{MAX} > 4\times$; which we consider to be lowering the system efficiency (hence the result is not plotted here).

When considering the wider system applicability, in this case, allowing extra packet transmissions means tolerating the resulting extra power consumed by the system, as well as potentially increased latency. Under a requirement of reducing power consumption, it would not be good to extend r_{MAX} too far. For others wishing to implement this system, testing the effect of changing r_{MAX} is relatively easy. The effect his highly application dependent, so this thesis only contains a general description of these effects.

Average Goodput observations

Additionally, we plot another graph showing different observations for not selecting $SS = 384$ as segment size. In Fig. 4.11, we plot an average goodput calculation for $r_{MAX} = 2$ to 5 for all segment size values. As can be seen, a segment size of 384 bytes yields very low goodput for $BER = 10^{-4}$ (bottom line of the graph), thus we are convinced that this is not worthy of use for such a condition (it would perform very similarly to $SS = 512$ bytes if we were to smoothen the graph by significantly increasing the number of repetitions in the simulation). It is also interesting to notice that for $SS = 256$, the goodput may actually be better for both $BER = 10^{-6}$ and 10^{-5} . However, we stick to our assumption not to use $SS = 256$, since $BER = 10^{-4}$ is the most reasonable and realistic condition to match our target BER operating range.

Moreover, referring to Equation (4.2), the goodput is linear with the number of correct transmissions. For channels with high error rate, the number of received packets will be low due to the discarded erroneous-packets. As this value is low, a lower goodput value is yielded, compared to the goodput in the channel with a low error rate.

4.6 Summary

To summarise this chapter, we have made use of the well-known TCP protocol to be used in our wireless transmission model. We started by giving an explanation based upon a comparison between classical wired and wireless transport protocols. Several research and improvement on boosting TCP performance have been elaborated – and from our point of view this brought us to reason there are still many positive opportunities for using and improving the TCP protocol in a wireless context. We started our TCP-based modelling by implementing a segmentation and reassembly concept combined with error control mechanisms. The purpose of this simulation

was fulfilled, i.e. to find the most effective size of segment during the transmission, and the number of maximum retransmissions allowed as we know that in an actual deployed system, suitable values need to be determined for both of these values; our simulations are able to show this. This value will be use for the rest of our simulation models, where we standardise $r_{MAX} = 4\times$ and fix the effective segment size at 128 bytes.

Chapter 5

Multi Parallel Independent Link Transmission Model Implementing Switching and Retries Scenario for Different Channel Conditions

5.1 Overview

In this chapter we further extend our model by combining multiple links into a multi-channel transmission model, i.e. a MIMO-like arrangement. We start with a brief overview of this technology and usefulness to underpin the relevance of this work to modern communications system, and then merge this approach with our TCP-based model to investigate cross-layer improvements.

We build two scenario models implementing different switching methods and retry scenarios for different channel conditions. The first one is called a dual parallel-link scenario model, where two Nodes A and B communicate over two sub-channels instead of one. The second scenario is a more interesting model which we name the

quad- parallel independent link model (Quad-PILM).

In addition to both models mentioned above, a novel switching method and retry scenario are proposed and evaluated for this class of communications system. By switching method, we refer to how each of the TCP payloads –after segmentation processing– are sent over the multi-link channels. Two switching methods are proposed, namely Segmented Window Switching (SWS) and Payload (PLD)-switching (will be explained later). In addition to that, to cope with erroneous segments, two retry scenarios are set up which show how re-transmissions will be carried out. These determine whether segment retries are sent over different links (RDL) or in a more arbitrary way, over a random link (RRL). More of these scenarios are observed later in the text.

For both models, we implement and evaluate both switching and retry scenarios, and the analysis behaviour is similar: we will investigate the system in the light of goodput under various channel differences. The analysis and discussions will be presented before we conclude this chapter.

5.2 Diversity and MIMO technology

5.2.1 Diversity

Wireless transmission involves signals propagating over multiple paths between transmitter and receiver. At the receiver, these multiple signals arrive with varying amplitude and phase which change in relation to each other over time, which is known as multipath fading. The fluctuating signals certainly yield a low signal to noise ratio (SNR) if they are simply summed together. This is unfortunate since achieving high spectral efficiency and quality is expected in future wireless data services [80].

Given the need for high spectral efficiency and transmission quality, there are

several techniques that are commonly used to improve the situation. The first is to introduce a higher modulation technique¹. Secondly, one can try to increase transmitted power. However this option is often not practical since it causes more interferences to other transmissions. Thus other techniques have been developed to mitigate these issues, in particular is called diversity techniques.

The principle of diversity is to provide the receiver with multiple versions of the same signal. This can then help to improve the performance of a wireless link so that the transmission error rate can be reduced as well as to stabilize an error-prone communication link. Several diversity modes are available to make wireless communications more robust even with varying channels. These are:

1. Time diversity.

In time diversity, a message is transmitted at different samples separated in time so that it undertakes uncorrelated fading processes. As for slow fading yielding a large coherence time, increasing the time gap between samples is not favoured. A study by [81] reported that this type of diversity is not bandwidth efficient due to the underlying redundancy.

2. Frequency diversity.

This form of diversity uses different frequency bands in the form of different channels. Similar to time diversity, this has to have enough of the spectrum to ensure the fading is uncorrelated at different frequencies. Thus this is also not always recommended as it requires more frequency bands to transmit the signal.

3. Space/spatial diversity.

Spatial diversity requires the use of multiple antennas located in different positions at either sender or receiver ends (or both) to take advantage of the different radio paths. In order to meet uncorrelated fading processes, the distance between the antennas should be far away from each other. This arrangement is

¹This will be explored more again later in the next chapter.

Figure 5.1: Receive and Transmit Diversity for the simplest scenario, where the figure shows SIMO (Single-Input Multiple-Output) 1×2 and MISO (Multiple-Input Single-Output) 2×1 arrangements respectively.

used as the basis for Multiple-Input Multiple-Output (MIMO) technology to improve the robustness of the transmission (spatial diversity) as well as increasing the data rate (spatial multiplexing). Practically, both methods can be used together or separately – depending on the channel conditions.

Spatial Diversity

The aim of spatial diversity is to make the transmission more robust by allowing redundant data to be conveyed over different physical transmission paths. This scenario is appealing in the case where bandwidth is a costly resource. Spatial diversity can be divided into receive and transmit diversity, as per the arrangement shown in Fig. 5.1.

Receive diversity utilises more antennas on the receiver side of the communication link. The basic idea is to have multiple signals experience different degrees of fading. This method is easier to implement since no special coding is needed. The multiple received signals are then combined into a single improved signal with the help of diversity combining techniques² (selection, switched, equal-gain, and maximal-ratio combining are some of the more common).

On the other hand, when there are more transmit than receive antennas, it is

²Please refer to [82] for more details on diversity combining techniques.

called transmit diversity. Compared to receive diversity, this method has the advantage that the redundancy codes (space-time coding³) takes place at the base station instead of the end user(s), thus allows less expensive and less complex implementation for a mobile transmitter.

Spatial Multiplexing

This form of MIMO, rather than making the transmission more robust, it is intending to increase the data rate (throughput/goodput). It utilises different paths to carry additional traffic as the data is divided into separate streams prior to being transmitted independently via separate antennas. This technique initially described in [84].

The next section will focus more on MIMO technology instead of the diversity topic, as the emphasis of this chapter is to adapt a MIMO-like concept to our cross-layer mitigation techniques.

5.2.2 MIMO technology

Multiple-input multiple-output (MIMO) is a radio antenna technology that can be represent as a link that utilizes multiple antennas at both sender and receiver ends to mitigate wireless transmission performance [85]. Usually a MIMO system consists of t -transmit and r -receive antennas. Using the same physical transmission channel, each of the antennas receives both direct and indirect signal components which may be intended or not respectively (illustrated in Fig. 5.2).

Multiple transmit and receive antennas are deployed in order to attain higher data rates compared to a conservative single-antenna architecture. This comparison has been studied in the work by [86] and [87], that MIMO systems can give remarkable spectral efficiency improvements over a single-input single-output (SISO) system.

³Proposed by Alamouti in [83].

Figure 5.2: Diagram of MIMO (Multiple-Input Multiple-Output) wireless transmission system.

The principle behind MIMO is that the signals arriving at the end of receive antenna(s) are combined in such a way that its data and error rates would be an enhancement as measured prior to transmitting from the sender antenna(s). MIMO technology can be viewed as the improvement of wireless transmission that in the 80s was called 'smart antennas' [88]. Nevertheless, its fundamental statistical nature goes a good way past the conventional smart antennas [89].

In MIMO, multiple antennas array at both end of the communication system are used to exploit the multipath propagation over the wireless channel, bringing about an improvement in system performance, increasing effective throughput (i.e. providing diversity gain) and improving transmission reliability (i.e. multiplexing gain).

Principally, its salient property is due to operation in a rich scattering environment (such as that provided by many man-made structures in towns, offices and so on). MIMO provides independent transmission paths (multichannels) from each transmit antenna to each receive antenna⁴. MIMO turns multipath propagation, which is considered a drawback of wireless transmission systems, into an advantage that can

⁴These transmission paths can be perceived as being independent channels of information – although in practice they will often be combined and processed together in the receiver before separate data streams emerge from logical channels, rather than the data streams being separated physically in space.

be used to increase the users data rate (initially reported by Foschini [86] [84]).

MIMO technology has emerged as one of the most promising research directions in the wireless community since it has become well-known for providing high data rate transmission without necessarily increasing bandwidth or transmit power of the antennas [6]. It has also become popular for its ability to provide higher spectral efficiencies (more bits per second) thus giving greater throughput or goodput to the end user, and improving link reliability (in this case, through reduced fading). Because of these advantageous features, MIMO is an appealing contender for the physical layer in 4G, as well as an important part of modern wireless communication standards such as IEEE 802.11n (Wi-Fi), 3GPP Long Term Evolution, WiMAX and also HSPA+.

Relevant work on MiMO technology

There are several relevant studies concerning MIMO technology and cross-layer TCP delivery approaches. For instance, [15] showed how data can be delivered over a MIMO link with the help of local link layer mitigation, without invoking TCP layer error control. In particular, it also demonstrated that practical MIMO links with given average BER, actually spend a substantial amount of time with individual sub-channels experiencing quite different levels of instantaneous BER.

In [90], the author proposed two different MIMO systems and claimed that in regard to application performance, physical layer approaches do not necessarily improve TCP throughput. However there are trade-offs possible between systems performance and SNR.

Other research in [91] suggested a new protocol for IEEE 802.11 MAC; the MIMA-MAC protocol to be combined with the spatial multiplexing capability of MIMO. Zhou et al. in [92] only considered MIMO systems limited with merely adaptive modulation (AM). Lin and Wong [93] proposed cross-layer design schemes, combining MIMO technology for different MAC protocols.

Recently, some authors [94] analysed power variation related to adaptive AM for space-time coded MIMO systems, while [95] proposed a framework of distributed MIMO ad-hoc networks for improving link capacity in regard to avoiding interferences among nodes.

5.3 Dual- Parallel Independent Link Model

This section will describe the simulation model already described, modified to adapt to a MIMO-like system arrangement. Note that we do not consider improving the MIMO system in the manner of spatial diversity / multiplexing to be our objective. The purpose of this thesis is to utilize a MIMO-like connectivity (that is increasingly common in implemented wireless systems) in a cross-layer approach to improve the reliability of wireless transmission.

We start with the model of a dual-antenna scenario instead of just a single antenna. Nodes X and Y form a single wireless communication system that has at least two physical sub-channels. Each sub-channel forms a separate digital link with slightly differing characteristics⁵. In this section we assume no more than two links (i.e. two subchannels between the transmitter and receiver, which may be physically separate links, or pseudo-channels such as eigen channels). The arrangement is similar to that of Fig. 4.4, except we have dual-link wireless transmission between Nodes X and Y.

The data transmission over each link will be subject to the time-varying and instantaneous BER and bandwidth experienced by that individual link. That is, for example, while the overall system BER may be fixed at 10^{-4} , each subchannel will

⁵This is the real-world experience of many research and commercial wireless systems [15], namely that where two streams of information can be conveyed by the physical layer, they may be characterised by their average BER, but in fact they vary instantaneously in time, and while one is ‘good’ the other may be ‘bad’. It is this variability that MIMO coding can exploit, and which the techniques discussed in this thesis can similarly exploit.

experience a BER that drifts up and down over time, with a long-term average of 10^{-4} (this will be explained more detail in the next section).

5.3.1 Channel difference

Motivation

Although real deployed systems would experience time-varying instantaneous BERs, our simulations – using Monte Carlo methods for bit errors – rely upon long-term averaging, so would not capture the effect of this variability. Thus we devise a method to artificially adjust BER experienced by each channel separately. In other words, we deliberately force one link to be slightly ‘bad’ and another to be slightly ‘good’ during the simulation, but ensure that their average BER remains as specified and that the total number of errors experienced is consistent with this average.

This is based on the some studies in multi-antenna system showing that channel characteristics are often unequal [96], for example in a two-channel system, one typically has higher BER than the other. It does not mean that one channel automatically improves when the other degrades, but simply that at any one time instant, one channel would naturally be better than the other and this is the case in practice.

There is no correlation here between bit errors in independent channels. Everything is calculated on a frame-by-frame basis, and over a single TCP packet transmission time and above, there is no correlation made between channels: they have entirely independent errors. The only constraint is that the long-term error rate is fixed and again this is what would happen in practice. In practice we would not know the precise long-term BER value until after a transmission run. But once the run had finished, we would characterise its performance by its average BER!

More on this, a practical test by [15] showed clearly that when exploring 2-channel statistic of a TR-STBC (time-reversal space-time block coding MISO) system,

it is found that (a) the distribution of BER of each channel is variably spread around the mean value, (b) it can be easily spotted that one channel is usually good while the other is usually bad in a real system⁶, and (c) the BERs value in the channels show certain differences, but their averaged performance is identical. This means that in a real deployed system, it would experience time-varying instantaneous BERs according to the statistical properties of the transmission over the particular wireless channel.

The above findings are the fundamental motivation in this thesis to use the BER statistic in modelling the multi-channel system. It is very normal in fact, for the channels to be slightly different than it is for them to always be exactly the same. And yet the vast majority of simulations tend to assume a fixed BER equally distributed between channels.

Modeling

Based on the premise above, we model the system by assuming both of the links have different channel conditions, i.e. one channel slightly worse or even better at same time. First, take in that for every B_T bits in a communication channel, we define an average BER target:

$$BER_{target} = \frac{1}{B_T} \quad (5.1)$$

We then introduce a BER control variable as $\delta_{(m)} = m(0.1B_T)$ to unbalance the links. This changes BER experienced by one particular link relative to the other link, with a channel difference coefficient, $m = 0..9$. For two channels A and B, we have:

$$BER_A = \frac{1}{B_T + \delta_{(m)}} \quad (5.2)$$

⁶It could be because of antenna placement, the impedance of RF feed wires, inefficiencies, or drift in the RF circuits at both transmit and receive end.

Table 5.1: Dual-link B_T distribution

	$m = 0$	$m = 1$		$m = 9$
Link A	1000	1100		1900
Link B	1000	900		100

and

$$BER_B = \frac{1}{B_T - \delta_{(m)}} \quad (5.3)$$

Combining (5.2) and (5.3) –here we take an example of $B_T = 1000$ – then we have the BER distribution value for dual-channels as given in Table 5.1. In all scenarios, it should be clear that the systems average BER (and total number of errors per unit time) remain constant. The simulation was run with the parameters as established previously, but now we fix the r_{MAX} value = 4.

5.3.2 Switching methods

In this section, two packet switching methods will be explored in detail. As to switching mechanism, the dual-link model will route packets down each link accordingly, based on two proposed methods. The first method is named Segmented Window Switching (SWS), and the second is Payload (PLD)-switching. The differences between the two methods are clearly illustrated in Fig. 5.3 which shows the example of two TCP data payloads that are split into four segments before being transmitted. The important difference is the sequence of segments that are transmitted, and the identity of the subchannel that they are transmitted over.

In the SWS scheme, the segments are transmitted over available links in a round-robin fashion. By default, the first segment ($a1$) will be sent over Link A. This will be followed by the subsequent segment ($b1$) on Link B, then will switch back again in a round-robin fashion until all the segments are through. The PLD scheme, by contrast, uses higher-layer knowledge to force each segmented window from one particular TCP

Figure 5.3: Switching methods illustrated with two TCP payloads each divided into four frames and transmitted on successive timeslots over two physical links.

payload to be conveyed over the same link. Once the current payload is sent through, then the link will switch for the next payload.

Fig. 5.3 is a simplified diagram – in practice, with many TCP packets being transmitted, there would not be any idle slots. In all simulations, we use equal bandwidth links, but they will differ slightly in instantaneous BER as mentioned previously.

5.3.3 Analysis of switching methods and channel difference model

The effect can be seen in Fig. 5.4 for both SWS and PLD methods for different average BERs. The extreme left of the plot, where there is no channel difference, simply shows the goodput available for averages BERs of 10^{-3} , 10^{-4} , and 10^{-5} respectively. Moving to the right, as one channel degrades and improves correspondingly (as shown in Table 5.1), we see this has a negative effect on goodput even though BER remains constant throughout.

At a BER of 10^{-3} , it can be seen that as channel difference increases, SWS starts to fail early when the maximum simulation time limit is reached (it is when the plotted line ends, showing no goodput at all). By contrast, the PLD method does not give up as easily as SWS does. The goodput is there, though very low. This

Figure 5.4: Dual-link model with different switching methods and channel difference.

shows that when the channel is inconsistent, none of the methods seems to be working optimally.

Moving to more moderate BER of 10^{-4} , in general, the tendency is similar. From consistent channel conditions moving to erratic ones, the goodput results of both methods are alike. Just as the channel gets worse, we see that PLD performs better, i.e. the goodput of the SWS method drops as the line ends while PLD gradually decreases in performance. Note that the operation of this simulation was for conditions of quite severe BER conditions.

For improved BER conditions of 10^{-5} , the inclination is similar for both SWS and PLD yet the goodput is still there. This can be explained that due to the good BER condition, both methods are working fine.

Overall, with poor average BER, SWS effectively corrupts almost all TCP packets (yet PLD allows half to pass); whereas with good average BERs, PLD concentrates errors on alternate TCP packets (yet SWS spreads the errors more evenly

To conclude this section, we can see that for the given scenario, PLD may be better for lower average BER in a dual link arrangement. This is consistent with other findings in [15], although the previous work did not take into account the full TCP/IP protocol, as the work reported in this thesis does. Most interestingly, we can see clearly that the ‘best’ arrangement for payload switching will change depending upon the overall average BER. For practical wireless linking systems, these findings indicate that it may be advantageous to incorporate different payload switching mechanisms that are selected based upon current operating parameters (for example: overall average error rate, channel condition, etc)

5.3.4 Retries scenario

We had successfully implemented the switching methods in the varying BER channels. However, after the analysis, we found that our system did not adjust very well as expected in terms of the erroneous segments. Basically there is no error control mechanism applied for this dual-links model. The transmission processes uses the ‘usual’ error control mechanism as typically applied for a single-link transmission, which is a retry and reACK mechanisms.

In this fashion, in order to improve our dual-links model, for each method we implemented and evaluated two different retry rules to handle segments received in error (or not received). The concept is shown diagrammatically in Fig. 5.5.

The first is where retries are re-transmitted on a different link (RDL) to the one over which they experienced an error. In other words, instead of re-sending the erroneous segment (a) over the same (top) link (i.e. the same link which exhibited the error which caused the re-transmit), the segment will instead be transmitted over the other link (bottom link). The subsequent segments (b and c) is kept in its default link. For PLD switching methods, this trend is similar. The erroneous segment will be re-sent over the other link. Things to note on this scenario are, since this always

Figure 5.5: Two retry scenario for the dual-link transmission model for both SWS and PLD switching mechanisms.

means that when a packet is sent twice (i.e. resent once), it utilises both the better and the worse link, the errors are spread, and overall error rate matches the average BER.

The second rule utilised a random link (RRL) to re-transmit erroneous segments. This may involve either re-using the previous link or using the other one (i.e. randomize), and basically allows the system to retransmit over whichever link is next available. As can be seen at the bottom of Fig. 5.5, the erroneous segments are ‘free’ to utilize any available link. By contrast to the former scenario, the original default behaviour was to always retransmit erroneous segments over the same link that had failed previously.

As this scenario will be used under TCP protocols, the details of the interaction with segments on the same link are:

a. independent.

Each segment is transmitted independently. For example whether segment (*a1*) is successfully received or not in particular link, it does not affect the subsequent segments except when it has reached the maximum retries value for that packet.

While a segment is in the transmission process, other segments are waiting in the available buffer (at Node X). Note that, the system does not overwhelm the memory buffer since the buffer is limited based on the number of the block sizes created.

- b. set apart from the sequence number (SN).

Node X would be able determine -for each block size- which link to transmit based on its sequence number. It uses a simple computation coding to make sure each block size has a fixed link-number - whether it would be transmitted into other or random link.

5.3.5 Analysis of switching methods and retries scenario under different channel conditions model

We then apply these concept into our simulation model. Again, the remaining simulation parameters are those defined previously, with the results and discussions as follows.

The graph in Fig. 5.6 presents the goodput of all methods with respect to increasing channel difference. Though difficult to examine since the condition was for good BER (and thus few errors), we can see that at low channel differences, performance of all schemes is identical. However as channel difference increases, the first retry scenario for both switching methods (S-RDL and P-RDL) prevails slightly compared to the other combinations. Since BER is good, there is only a very slight performance improvement for both RDL (retransmit over the other link) and RRL (retransmit over random link) over the default behaviour as channel difference increases.

However for moderate to worse BER, as seen in Fig. 5.7, we see that besides S-RDL, P-RDL also performs slightly better under same conditions as channel difference

Figure 5.6: Comparison of all dual-link performance methods for different channel conditions, $BFR = 10^{-5}$.

Figure 5.7: Comparison of all dual-link performance methods for different channel conditions, $BER = 10^{-4}$.

increases. The reason is that P-RDL, in good BER links, will preserve the transmission in that link ensuring a complete and good, TCP frame delivery. By contrast, S-RDL will retransmit and pass the segmented frame to the other link (round robin fashion), thus slightly reducing goodput – since the retransmitted segment is considered failed, therefore this has an effect on the goodput calculation.

Figure 5.8: Link A consistently get worse, $\text{BER} = 10^{-5}$.

From these results, we conclude that the RDL method is advantageous for systems with poorer BER than RRL. Overall, the P-RDL mechanism performs best.

5.3.6 Another possible channel scenario

To assess another possibility, we added another scenario to the investigations. This time, Link A is forced to consistently get worse (instead of getting better as used in all previous simulations), while the remaining links improve. The results are shown in Fig. 5.8 and Fig. 5.9.

In fact, this result shows that the new scenario yields S-RRL as the best approach, slightly outperforming P-RDL. For good BER, SWS switching prevails for both retries scenario (S-RRL and S-RDL), while for worse BER, the RRL scenario prevails for both switching methods (S-RRL and P-RRL). This is a very interesting

Figure 5.9: Link A consistently get worse, $\text{BER} = 10^{-4}$.

finding, showing the particular strengths of the two adjustment methods tend to be complimentary (i.e. one relates poor BER while another relates to good BER). Of course, *the most important fact to remember is that the methods proposed here do in fact yield an improvement*. The precise method which is best in any particular scenario may differ.

So these two findings suggest that different conditions will yield different results. This is one reason why the methods are clearly and openly explained in this thesis, and are very clearly outlined in the papers we have written about this: it is precisely so that other researchers can implement the method for their own systems, past, present and future (with the motivation that the system has been shown to work well on several test scenarios). We consider this to be a much more sensible idea than to have to test hundreds of possible alternative systems and scenarios.

Regarding the following experiments, since they continue with similar conditions

to the original experiments (i.e. maintaining the original situation of Link A getting better), the assumption remains, that P-RDL is the best approach for this scenario and is hence adopted into the future experiments (unless otherwise stated in those experiments).

5.4 Quad- Parallel Independent Link Model

5.4.1 Altering-Link Modelling

In this section, we extend the simulation to a more interesting MIMO model named the Quad-Parallel Independent Link Model (Quad-PILM). The motivation, methodology, packetization methods, retry alternatives and evaluation criteria remain the same, along with the previous simulation conditions, except this time we explore performance over four parallel links, and that the channel degradation is now more one-sided.

Since we cannot use the a symmetrical channel difference between two links, here we adopt a 1–3 link balancing comparison. In this case, one link (Link A) is controlled in terms of its BER, meanwhile the other three links progressively degrade at 1/3 of the rate. By doing this, we maintain overall BER (i.e. total proportion of bit errors) in the system to be equal for all channel difference scenarios.

Similarly to what has been implemented for the dual-link model, this Quad-PILM system implements the same switching and retransmission scenarios. The arrangement can be seen in Fig. 5.10.

As discussed in the behaviour described previously, for S-RDL in subfigure (i) and P-RDL in subfigure (ii), when a segment fails, it will be re-sent to another one of three remaining links. Subfigures (iii) and (iv) involve resending over a random link, which could be the same link that just failed (although the probability of this

Figure 5.10: Quad-link switching methods and retries scenarios showing (i) S-RDL, (ii) P-RDL, (iii) S-RRL, (iv) P-RRL

happening is obviously lower than it is for the dual link system).

Referring to (5.2) and (5.3), the Quad-PILM simulation runs with the BER distribution for Link A, given as:

$$BER_A(n) = \frac{1}{nB_T} \quad (5.4)$$

and for the other three links will be :

$$BER_{B,C,D}(n) = \frac{1}{LB_T - BER_A(n)} \quad (5.5)$$

with L and n indicating the total transmission links and the channel difference distribution for quad-link model, respectively. The BER distribution can be seen in Table 5.2.

Table 5.2: Quad-link B_T distribution

	$n = 0.8$	$n = 0.9$	$n = 1.0$	
Link A	800	900	1000	
Link B	1067	1033	1000	
Link C	1067	1033	1000	
Link D	1067	1033	1000

Figure 5.11: Quad-PILM: SWS-switching of all retry mechanisms for $\text{BER} = 10^{-3}$.

5.4.2 Discussion-1

We again examine performance to highlight the difference due to changing switching method in PILM for poor BER conditions. Fig. 5.11, plotting goodput against channel difference distribution, clearly shows that for Quad-PILM, methods S-RDL and S-RRL both finally outperform the original SWS as the value exceeds 1 and beyond. In this case, channel difference needs to be explored in both a negative and a positive direction because it is no longer necessarily symmetrical (as it was for the dual-PILM case).

When the distribution value less than⁷ 1 –the left part of the figure– the diversities in the four links are significant, yielding less goodput compared to the ideal and stable channel conditions where the value equals 1. Similarly, the trend should be similar when the distribution value exceeds 1. However as the retry scenario works perfectly then we notice the goodput is improving instead of decreasing.

⁷Note that for the distribution value $\neq 1$, it means that BER differences exist between link A and others.

Figure 5.12: Quad-PILM: PLD-switching of all retry mechanisms for $\text{BER} = 10^{-3}$.

Between the improvements of the switching methods, we found that when the system delivers the retry frames into a random link (RRL), it performs slightly worse than when retries are delivered into any three other links apart from the erroneous one (RDL). This can be explained as being due to the latter not re-transmitting the frame into the current failed link; while in RRL, there is always a chance the frame is delivered again over the failed link.

This trend also applies to the payload-switching methods (see Fig 5.12). When the distribution value > 1 , the original PLD is the worst compared to all of the improved PLDs (P-RDL and P-RRL), showing that for PLD switching method, the retry mechanism also works very well by giving goodput improvement. The RDL scenario will simply prevent the re-transmitting frame from being delivered into the same susceptible link, and thus will slightly surpasses the RRL scenario.

Figure 5.13: Quad-PILM altering link: SWS-switching of all retry mechanisms for $\text{BER} = 10^{-3}$.

5.4.3 Discussion-2

To gain confidence both in the switching methods and retry scenarios of our Quad-PILM simulation, we ran another experiment to see the outcome if the controlled link is changed⁸ – instead of Link A, now we performed the control alteration at Link B.

The simulation is run for both SWS and PLD switching mechanisms, combining both links alternations. The results for the alteration for SWS-switching are given in Fig. 5.13, while those for PLD switching are shown in Fig. 5.14. Taken as a whole, it can be seen that, regardless which link is used for alteration, the goodput results are similar. This removed one potential complication in the experimental results.

⁸In our simulations we always assume Link A to be the default starting transmission channel for transmission sequences, and it was also the channel that was controlled to obtain the channel difference results.

Figure 5.14: Quad-PILM altering link: PLD-switching of all retry mechanisms for $\text{BER} = 10^{-3}$.

5.4.4 Discussion-3: Randomising channel's BER

To make the simulation conditions more realistic, two additional studies were performed in which the channels' BER is randomized, i.e. to let other links experience BER modification. We took account into two more setups: (1) where half of the channels have a BER that is improving and the others are degrading, and (2) where one channel is improving, one is degrading and the remainder are fixed. The results are shown in Fig. 5.15, where the subfigures (a), (c), (e) and (b), (d), (f) show the SWS and PLD switching methods respectively.

The first row of the figure shows the results of the original scenario (i.e. only one-link is getting better) which will be used for the comparison to the setups no. [1] (Fig. 5.15(c) and Fig. 5.15(d)) and no. [2] (Fig. 5.15(e) and Fig. 5.15(f)). As

(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.15: Randomized channel BER.

a whole, the results indicate that, even when the channel property is randomized; it does not invalidate the conclusions from the original scenario in this section. It is very obvious that the original SWS and PLD methods always fall short compared to the enhanced retry scenarios (RDL and RRL). And again, the inclination of the graph shows that RDL outperforms RRL. This, combining these two findings, P-RDL is still seen to be the best mechanism for the given test scenarios.

5.5 System Complexity

5.5.1 Computational and Deployment Complexity

There are several important aspects related to the computation at both transmission nodes. First is the current-link ID information at the transmitter. This identification is crucial for determining the switching decision for both scenarios. Practically, this ID can be used in conjunction with the sequence number (as the TCP protocol does) but with some minor additional logic at the controller node. The system should also make sure that this sequence number is well utilised. This is due to the fact that the frame reconstruction process will fail if this number is not received correctly [79]. A failure in frame reconstruction is extremely significant in that it potentially invalidates a large number of segments.

In terms of real-world system deployment, to fully utilise the system, it is preferable that the condition of each transmission channel (i.e. in terms of error rate statistics, including their variation) is tracked. In this way, the system controller can select one of the operating regimes appropriately depending upon the current operating environment of the system. However, for deployment where the above mentioned condition is infeasible, then an estimation of packet error rate (PER) over each link can be exploited – this would involve virtually no significant additional computational complexity in the system controller, just a simple counter to accumulate error events

per channel.

5.5.2 Latency and Buffer requirements

In general, additional latency over a baseline system is closely related to several causes. In the paper, it is obviously affected by the protocols employed. Starting with the SAR-ARQ protocol (retransmission scheme), followed by the TCP mechanism (mainly congestion window and fast retransmit and recovery), and lastly when an AM adjustment is employed. Thus each of these may increase the system latency measured as the time delay between a data bit entering the transmitter and exiting the receiver. The effect of this latency, noticeably, will be in the performance of the system in terms of goodput. Subsequently, though the paper did not use latency as a main metric in the evaluation, it can be observed as one of the major factors influencing the goodput performance of the system.

There is also a secondary latency to consider, and that is the time delay between the channel conditions changing and the system adapting. This is entirely implementation-dependent, but as the adaptation mechanism relies upon information fed back from the receiver to the transmitter, it is affected by packet buffering in the reverse direction. This type of latency effect has been explored in several papers, such as [16] which uses a prediction scheme to mitigate against delay-induced performance degradation.

Any SAR-ARQ system will require both transmit and receive buffering. The obvious additional requirement here is that if a MIMO link is decomposed into N separate channels, then these buffers increase by a factor of N for the separate links. In practice the transmit and receive buffer sizes need only exceed the maximum length of one TCP packet. Any buffers to the input and output TCP data streams visible to higher layers will remain unchanged compared to a baseline SAR-ARQ implementation.

5.6 Summary

In this chapter we have constructed two models, i.e. dual- and quad- parallel link arrangement of a wireless system, as the implementation of MIMO-like technology. We implemented our TCP-based model into this MIMO arrangement to test and evaluate its performance.

Along with the models, two switching mechanisms named segmented window switching (SWS) and payload (PLD) methods are employed. In conjunction with these, two different scenario rules were also implemented together, one in which the retransmitted packets are sent on a different link (RDL) and the other rule in which packets are sent over a random link (RRL). All of the switching and retry circumstances are implemented due to the observation of varying BER channel conditions.

Based upon the results and discussion above, we show that retry mechanism have given significant improvement in the given both switching conditions in the quad-PILM case. Also demonstrated for the scenarios tested, the payload-switching with the RDL retry scenario (P-RDL) is the best approach for the given conditions, compared to RRL.

Lastly, combining the switching mechanism and retries scenarios in the quad-parallel link arrangement, we discussed several possibilities in our model. From the results, we conclude this chapter by noting that the payload-RDL (P-RDL) arrangement is best for the poor BER scenarios explored, and it will be adopted for the remainder of this thesis.

Chapter 6

Quad-PILM with Adaptive Modulation

6.1 Overview

This chapter will discuss a cross-layer AM (Adaptive Modulation) switching concept in relation with our existing model as explained in the previous chapter. First, we briefly introduce relevant research on AM by discussing several related papers that match our aims in this thesis. Moving to the modelling, we address a novel simulation scheme that combines the P-RDL scenario of Quad-PILM TCP-based model (i.e. the best-performing output from the previous chapter) with the use of AM for a point-to-point MIMO wireless system. In total, we will develop four scenarios, gradually enhancing the performance of the model towards the end. For each model, a separate analysis and discussion is presented before we conclude this chapter.

6.2 Adaptive Modulation

Achieving both high data rate and spectral efficiency has long been a goal for future communication systems. The using of MIMO technology has been discussed in the previous chapter. MIMO is attractive it because can reduce the effects of multipath fading by providing diversity gain and improving reliability without (necessarily) incurring additional bandwidth and power costs.

Besides MIMO, another solution to multipath fading effects is to switch the transmission signal characteristics, i.e. adapting signal parameters such as coding rate, transmit power, or even to swap transmit antennas. Link Adaptation or Adaptive Modulation [97] describe the ability to change code rate or transmission power as a response to channel conditions. In such schemes, during transmission, the power of the transmitted signal is held constant over a frame interval, and is set according to past channel conditions, or predicted current conditions. The conditions measured to trigger switches can include SNR, BER, received signal strength, channel gains or other information. Several studies have been shown that Adaptive Modulation (AM) can improve unbalanced conditions between BER and spectral efficiency [98] [99] [100].

The AM method indicates the flexibility of matching channel properties, i.e. coding, modulation, and signal and protocol parameters, to the conditions of the transmission link. Or in other words, the signals modulation and coding format is altered to match the channel conditions or received-signal quality. Thus it possesses a dynamic nature in which the signal and protocol parameters are adapting to the varying radio link conditions. Over fading channels, such systems can demonstrate great improvements in performance, and are able to enhance the throughput/goodput of channels experiencing high levels of BER. For example, in flat fading channels, AM defines a set of signal-to-noise-ratio (SNR) thresholds which will be used to select the most appropriate coding or modulation scheme, depending on the SNR condition.

In the future, it is likely that many communications systems will utilize these methods. Some possible scenarios are, for example, in MIMO systems where the switching between antennas happens while adjusting transmission parameters as the channel characteristics change. In 2.5G and 3G mobile wireless standards: UMTS does an adaptation every 2ms, while EDGE uses a rate adaptation algorithm MCS (modulation and coding scheme) to adjust the radio channel quality in order to improve overall bit rate and robustness.

Regularly, AM systems –which adapt transmit characteristics on the basis of what has been received– require some form of channel state information (CSI) at the sender (transmitter). CSI could be acquired in time division duplex (TDD) systems by assuming the channel is balanced between the sender and receiver in both ways (i.e. bidirectional), or by obtaining CSI information at the receiver and then feeding it back to the transmitter¹.

6.2.1 Previous work on AM

AM has been investigated widely as a solution to mitigate against channel propagation errors, interference scenarios, and changing traffic or data rate requirements. It usually involves a switching mechanism rather than a continuous adaptation, with its aim to enhance the spectral efficiency of wireless systems while assuring QoS requirements.

Some of the early work on this topic was initiated by Pottie [101] and [102]. It was followed by Goldsmith and Varaiya [98] who investigated the capacity of an ‘adaptive’ system over flat-fading channels when perfect CSI is available at both sender and receiver. Then, in [103] and [104], they showed that the capacity of a channel can be reached using a certain rate / power-adaptation scheme in which the information bits is continuously updated according to the channel condition at any given time. They have also demonstrated that in a rate-adaptive system, a constant transmit

¹This scheme is not discussed in this thesis, but can be found in [16].

power has similarity with the one with both rate and power optimally selected.

Since then, the technique has been investigated widely, with the main purpose of matching transmission rates to varying channel conditions, for example in [105] [106] [107] [108] [109] [110] [111] [112], and even for the realistic scenario where only outdated channel information is available [113], erroneous [114], or even in multiple-input-multiple-output orthogonal space-time block code (OSTBC) environment [115].

In relation to the various implementation layers, much research has also been conducted. At the physical layer, though this may not be either practical or optimal in all situations, AM is usually designed independently from other layers. This is based on the premise that data packets are always available at the sender, which does not concern itself with buffer space or queuing needs. Practically, wireless packet networks can have random data arrival times and orders, thus making the queuing behaviour dynamic. In such situations it may not be considered practical for AM to operate in stand alone mode. This is what was analyzed by Liu et al. [116] who proposed an interaction of queuing at the data link with AM at the physical layer.

This assumption that the packets in the queue are not deadline constrained, however, does not reflect for some real time application which are constrained to a delay bound of 40-90ms, as claimed by Harsini and Lahouti [117]. For this kind of application (for example video-conferencing), the packet will be dropped once it passes a certain deadline. And this was what the authors considered in their work, which is to analyze the joint effects of packet queuing and AM. Partially, it was a good approach, as long as the queued-buffer size is being taken into careful consideration. Otherwise, once the buffer overflows, it actually will yield another packet loss.

More related studies regarding cross-layer design approaches have also been carried out. For example, in delay-tolerant applications, a combination of AM with truncated ARQ protocol and limited number of retransmissions has been considered in [118]. AM parameters were used to make sure that the PER be lower than a

threshold defined by the related application. This work then was improved by [119] by proposing a new approach of AM referred to as A-AMC (Agressive AMC) by using more spectrally-efficient transmission modes for the first transmission before switching to the more conservative transmission mode for packet retransmissions.

Related to the TCP layer, one novel concept named TCP “Adaptive-Selection” was proposed by [120]. Basically it extends the concept of AM to the transport layer. Encouraging preliminary result of a simple topology was claimed though there still exists the possibility of extending and discussing more implementations. Another work by [121] showed a queuing model at the link layer model by combining AM and ARQ to estimate the throughput performance in TCP layer. The model was able to achieve a suitable SNR threshold of the different transmission modes in relation to ARQ at the link layer, thus the TCP throughput can be maximized. More research regarding TCP performance and AM also include [122], [123], [124], and [125].

6.3 Modelling

Incorporating the concept of AM to improve our system performance, we address a novel simulation scheme that makes use the P-RDL scenario and Quad-PILM TCP-based point-to-point nodes of our previous simulations with an Adaptive Modulation capability. The idea is based on combining and analysing the different BER values over a fixed link in which QAM modulation is used, and to provide the ability to adjust the QAM level. The exact values are obviously system dependent, but for comparison purposes, our simulations use BER values as reported by [2], shown in Table 6.1. We also assign an appropriate channel data rate for each QAM setting.

In this system, we allow a channel that experiences persistent errors (i.e. the one that is likely to be the worst channel in a set of MIMO channels) to reduce its modulation based upon some packet error requirement. Thus channels will independently

Table 6.1: BER value for different QAM modulation at SNR = 15 dB, as reported in [2].

QAM	Data rate (Kbps)	BER
128	256	1.10^{-2}
64	128	8.10^{-4}
32	64	$1,5.10^{-4}$
16	32	1.10^{-7}

adapt their modulation format. When the adjustment is made, the BER experienced by that link will naturally change accordingly, as will the data rate over the link. For example, halving the data rate for each downwards step in modulation, and doubling the data rate for each upward step (although please note that the simulations reported here only include downward steps since the system starts with the highest QAM level selected for each link: 128).

6.4 Quad-PILM without Adaptive Modulation

Initially, for benchmarking purposes, we first plot a normal result for each QAM modulation, without any AM adjustment. We start with the highest level of QAM modulation that specifies the particular datarate. For each simulation run, we plot the goodput, and repeat for several different values of r_{MAX} (the parameters are those reported in Section 4). Results are plotted in Fig. 6.1.

We can see that the result is as expected: a higher data rate yields higher goodput. It can also be seen that for each QAM level increment, the goodput also improves – this happens for both r_{MAX} values.

In terms of the retry value, the bigger the value (i.e. $6\times$) the more segments are re-sent on error, and thus the higher probability that they will be received correctly – leading to improved goodput over $4\times$.

Since the system we are simulating is a Quad-PILM link, we can also explore the

Figure 6.1: Quad-PILM with no-AM, $r_{MAX} = 4\times$ and $6\times$.

effect of channel difference as normal. Naturally, the P-RDL mechanisms allow the goodput to improve by exploiting degree of channel difference. This graph has been plotted as a baseline, and also to validate the underlying simulation model which has now been adapted to utilise the AM concept (although no AM is performed in this graph – that will be explored in the next section).

6.5 Quad-PILM with Adaptive Modulation

6.5.1 AM-scenario

To assist our model in implementing AM, we now consider an additional parameter called NAK , which shows the total number of retries that have occurred at the sender (or from the receiver's perspective, it counts the total number of un-successfully received segments). The notion here is that, when the sender NAK value increases during data transmission, it means that the communication link is poor. The cumulative value (in our system we call it m_{NAK}) will trigger the system to perform an AM adjustment, i.e. reduce the modulation, and consequently dropping the bandwidth

Figure 6.2: The Quad-PILM system with AM triggers a modulation adjustment when m_{NAK} erroneous resends have occurred.

(data rate) by one step, which will in turn change the measured BER accordingly.

More detail of the process is shown in the flowchart of Fig. 6.2. Simply when one of the link fails, the parameter NAK will be incremented. For in case it has reached m_{NAK} , the adjustment will be performed according to the condition given in Table 6.1, and the accumulated NAK counter for the link will be reset. If m_{NAK} has not been reached, the system will simply continue with the normal retry mechanism by resending the failed segment (up to r_{MAX} times).

However, as we progress with this scenario, we found the system actually did not show ‘fairness’ to all channels. As concluded in Section 5, our system adapts the retry scenarios based on random chance, i.e. for each retransmitted frame, it chooses randomly between the other 3 links (RDL-scenario). Unfortunately, with this scenario, we found that the system always takes into account that the first adjustment occurs over Link A. In other words, it assumes that Link A is the source of the error which is not always true in practice: although it is more likely that Link A is erroneous, because the link errors are random, there will naturally be periods when one or more of the other links (although having better SNR) actually experiences more errors. With this condition, the system will perform two unwanted actions: (1) forcing Link

Figure 6.3: AM scenario with LV parameter— with m_{NAK} is counted for each link (left) and collectively at node base (right).

A to always downgrade its data rate when errors occur elsewhere and (2) actually, by doing this we did not always fix the problematic link.

Up to this point, though AM adjustment has been implemented in the system, it does not perform as desired. Thus we decide to upgrade and improve our system by incorporating a concept of ‘fairness’ among channels.

6.5.2 AM-scenario with link value parameter

To mitigate the issues in the previous section, we introduce another parameter, link value (LV), which tracks the performance of each link individually in terms of PER. For a successfully transmitted segment over a particular link, we increment its value, and decrement it for an un-successful one. As always, our aim for the system switching rule is simplicity. The NAK parameter still be used as an indicator of when to perform an AM adjustment, but will no longer specify which link to adjust. Therefore, this system is now performing longer term global performance tracking, with a shorter-term local per-channel performance indicator.

This arrangement is shown on the left of Fig. 6.3 where the flowchart shows

that once the sender becomes aware that one of the links has a problem (un-ACKed segment), it will increment both LV and NAK values for that particular link. As transmission continues, if NAK reaches the links m_{NAK} limit, then the system will inspect the LV s value for each link to find out which is the worst one. The AM adjustment will be performed on that problematic link then transmission will continue.

With this, whilst meaning that occasionally the incorrect link will be adjusted, it is a simple, portable and pragmatic method of tracking adjustments. The alternative, shown on the right hand side, will be explained in the next section.

Discussions

We run the simulation, varying the value of m_{NAK} and comparing against the un-adjusted (i.e. no AM) scheme, with the results as plotted in Fig. 6.4. In fact, we found that with $m_{NAK} \geq 3$, no adjustment occurs throughout the duration of each simulation, while for $m_{NAK} = 2$ the curve shows improvement over the others.

For the three bottom line of the graphs, it seems that the two of the AM scenarios yield similar results with no AM scenario. In order to explain this, consider that the

Figure 6.4: No-AM versus AM-LV₁ with varying m_{NAK} .

default value for maximum retries, r_{MAX} (as used in the previous section) and m_{NAK} value is set to 4 and 3 respectively. Remember that this m_{NAK} value applies for each link, thus the system will have a particular value (for extreme comparisons, it is 12) to compete with the maximum retry value, i.e. 4. At this case, it is very possible that no adjustment will happen, since most of the time the system will discard the segments (and automatically the payload) once it reached maximum retries value, r_{MAX} . This is most likely to happen before the maximum NAK at a particular node is reached. As the result, we see that performance is actually similar to having no AM scheme at all. Thus this behaviour will affect any scheme with $m_{NAK} \geq r_{MAX}$.

On the other way around, for m_{NAK} equal to 2, a significant improvement on goodput can be noticed. This is the opposite of the previous explanation that, as the value of m_{NAK} is set to 2, the AM-scenario is actually working as a very early ‘filter’ to force the system to perform the adjustment before significant link retries occur. As the consequences, when any of the links experience error, the adjustment happens as soon as the NAK value reached 2, thus increasing the overall chance that segments arrive safely at the receiver.

However, we found that $m_{NAK} = 2$ may be too premature to force an adjustment (it is acting upon insufficient information to make a guaranteed correct choice), although significant goodput is achieved in the simulation. We thus consider upgrading the system to counter this problem as will be shown in the next section. However the system as presented here has demonstrated that AM adjustment using a cross-layer methodology (i.e. counting PER) can improve performance.

6.5.3 Updated link value AM-scenario

Now, instead of counting the total NAK s at each link, which we found to be unrealistic as no adjustment happens when $m_{NAK} > 2$, we develop another scenario to count the global NAK value at the transmission node itself – we name this AM-LV₂. The flow

Figure 6.5: No-AM vs. AM-LV₂ scenario, with varying m_{NAK} .

of the updated method can be seen in Fig. 6.3 (right hand side).

Here, every time any link experiences an error, a global NAK counter (instead of local NAK for each link in the previous scenario) will be incremented. Similarly, LV will be updated accordingly, until the base NAK value reaches m_{NAK} . At this point, the system will decide which link has the worst value by comparing LV values, and will subsequently perform the adjustment on that link with the worst LV . In the case of links have the same LV values, we simply adjust the link that performed worst the last time.

Discussions

Again, we compare this to the no adjustment (no-AM) scheme, with the results shown in Fig. 6.5. First, we see a significant goodput improvement for all m_{NAK} values compared to no-AM scheme. Here it shows that AM-LV₂ works better under the tested conditions (i.e. for all m_{NAK} values) than the previous AM scenario; with the adjustment scheme being well utilised.

For m_{NAK} is set to 4, it can be seen that for moderate channel differences (

in between 0.9 and 1.2) –since all the channels have similar BER– the probability of adjustment is very low, giving just a fair goodput. This is as expected since the channels have similar BER and little switching adjustment is needed.

It is appealing to notice that for more extreme channel differences, i.e. when its distribution is under 0.9 and above 1.2, the goodput improves significantly compared to the moderate conditions, which indicates that the AM-LV₂ scenario works perfectly. We argue this because when the channel difference is large, the adjustment will happen more often – and this will increase the possibility of the frames being received correctly at the receiver.

A noticeable improvement in total goodput is shown in the graph, showing a step change in performance due to a thresholding effect on the experienced PER – and this indicates that the system is well tuned. The system as presented here has demonstrated that AM adjustment using a cross-layer methodology (i.e. counting PER) can improve performance.

Close Observation on the channel distribution transition

As we refer to Fig. 6.5, there are two ‘rapid’ transitions both on the distribution, which is happening in between 0.8 and 0.9 as well as 1.2 and 1.3. We try to observe more details about this transition, by plotting the goodput in Fig. 6.6 and Fig. 6.7. As shown in both figures, the goodput results are simply slightly changing as the channel differs.

Close Observation of m_{NAK} value

Looking closely at the different m_{NAK} values plot, we see more evidence that for large m_{NAK} , the system performs well using the AM-LV₂ scheme. Depicted in Fig. 6.8, it can be seen that bigger m_{NAK} values obviously give the system an opportunity to do more error recovery before adjustments happen. It shows that for $m_{NAK} = 0$, the

Figure 6.6: AM-LV₂ scheme for channel distribution between 0.8 and 0.9.

goodput is achieved at a minimum level – in contrast to any m_{NAK} value larger than 0. Note that the recovery will not affect the goodput much compared to halving the data rate.

6.5.4 Summary

In this chapter we presented a P-RDL based quad-parallel independent transmission link model (Quad-PILM) combining the segment switching and retry mechanisms with a simple adaptive modulation scheme operating on individual subchannel links.

We briefly discussed AM following the reason of implementing this method in our model. We started the simulation without AM scenario, by varying the maximum retransmission parameter to validate the underlying simulation model. Then

Figure 6.7: AM-LV₂ scheme for channel distribution between 1.2 and 1.3.

Figure 6.8: AM-LV₂ scheme with different m_{NAK} value.

we followed by developing another three AM scenarios to gradually improve overall goodput for those quad-links –this was done using these pragmatic and low complexity methods. As shown in the last reported model, we conclude that both in moderate and extreme channel conditions, the system performance improves through the use of AM adjustment. In particular, significant goodput was achieved for extreme link differences in the multiple antenna systems deployment.

Chapter 7

Conclusion and Future Work

In this final chapter, we present a conclusion of this thesis by underlining some important contributions of the work. In the last part, we discuss some future work and research to continue this topic.

7.1 Conclusion

This thesis focuses on cross-layer design solution for future multi channel wireless systems, in particular for simple wireless sensor systems that utilise TCP/IP, by implementing a novel mechanism of packet switching and retry scenario over different sub-channels. It requires that the underlying radio system –either using multiple physical channels or MIMO subchannels– presents those different subchannels separately to the higher layer. It exploits the known phenomenon that the instantaneous BER experienced by different subchannels can be significantly different from one time instant (i.e. packet) to the next in a real radio system.

Each chapter of this thesis started with an overview of the related topics. Following that, they run through literature surveys which include related research work regarding the issues under investigation. For some proposed methods, where possible,

a comparison is presented against the theoretical concepts. Following the construction of models and simulations, deeper discussion and observations were made prior to concluding the chapter.

In summary, the work of this thesis is separated into several stages:

Stage 1

At the beginning, the thesis reviewed and investigated several simulation tools to be used for building a model of the system under investigation. The motivation was simply to very cautiously decide the right simulation tool to be adopted based upon a number of implicit requirements (which would impact future EOS researchers, not just the work described in this thesis). Since there are many potential tools available, careful judgement must be made before fixing this decision. Details were given, particularly in comparing the three most famous network simulation tools which are ns-2, OPNET, and OMNeT++.

The decision to select OMNeT++ was that:

- a. OMNeT++ is found worthy to be selected as a tool for our models in terms of budget (open-source software), ease of debugging or use (easy GUI for non-specialist users), and availability of supporting protocol models. We also published this analysis as a book chapter [126].
- b. OMNeT++ consistency was confirmed by building a simulation model of lost and received packets with respect to bit and packet error rates of a known baseline system. We also compared the results against the theoretical concepts in order to verify correctness.

Stage 2

In the second stage, two concepts were defined that are very important in this thesis, which are segmentation and reassembly (SAR) and the introduction of error control mechanisms inspired by TCP protocol, but adapted for use over wireless links.

SAR concepts were developed under the consideration that it was necessary to make an efficient transmission process given the error characteristics of the links that are likely to be found in the field. This is due to the fact that it is not generally possible to send entire IP packets across a lossy wireless network without experiencing significant PER. Related with this, a block size parameter was defined which determines the length of segmented packets transmitted over the air, appended with a one-byte sequence number.

In terms of error control mechanisms, these make use of the common principle of retry and re-acknowledgement, which is used in conjunction with a timeout. Practically, the mechanisms were designed to ensure the segments were sent and received properly at each node.

For this stage, the major findings are:

- a. By simulating the concept of point-to-point wireless transmission utilizing SAR and error control mechanisms, it was found that (1) packet error rate percentage is directly proportional to the BER as expected, (2) the bigger value of parameter block size yields a higher percentage of PER, and lastly (3) a careful choice of block size is encouraged since a small block size implies a greater overhead in terms of segment header and ACK traffic.
- b. In terms of error control mechanism, under poor BER conditions, more segments need to be re-transmitted. Meanwhile, increasing block size does not automatically reduce the number of retries and reACKs.

- c. When increasing the retries and reACK values: (1) it is found that the bigger of these values gives a higher probability of segments arriving at the receiver, thus the PER percentage is reduced. (2) total bits transmitted over the air also increases (also confirmed by looking at the number of retries and reACKs).
- d. In terms of bit transmission efficiency, the system showed that it is inversely proportional to the block size for a noisy channel. The bigger the block size, the higher the possibility of the segment being lost, thus decreasing its efficiency.
- e. A deployment-specific trade-off needs to be considered in terms of block size and bit transmission efficiency, since the simulation showed that a poor choice of block size can affect the system efficiency greatly.

Stage 3

The third stage firstly adopted the TCP protocol as the basis of the ongoing simulation models. This began with discussions relating to TCP and its use in wireless systems, then combined this base model by implementing SAR concepts combined with error control mechanisms as have been built in the previous stage.

Second, the thesis considered the most effective size of the segment being transferred over the air and the number of the best maximum retries. At this stage the motivation was to have the exact analysis of the most effective size of these parameters in the expanding TCP model.

After developing several scenarios run, for the important results in this phase, the optimal maximum retries value was standardised at 4, and the effective segment size fixed to 128 bytes.

Stage 4

This stage shifted focus to a MIMO-like model by building up two and then four parallel links in an arrangement named the dual- and later the quad-parallel independent link model (PILM).

Together with the models, two novel methods were proposed. The first one is the switching mechanisms named segmented window switching (SWS) and payload (PLD) methods. Along with this, two different retries rules were envisaged, implemented and analysed. Switching methods indicate how the packet is switched during the transmission over the links, while the retry scenario affects where retransmitted packets are re-sent over a different link (RDL) to the original transmission or over a random link (RRL). All of these circumstances are observed in the varying BER channel conditions in the light of goodput.

For this phase, the major verdicts are given as follow:

- a. It was found that in terms of switching methods, PLD was able to maintain the system performance as channel difference increases, compared to the other method (SWS), for the scenarios explored.
- b. In terms of retries, it was shown that packet retransmission over a different link (RDL) provides better performance than the random alternative (RRL).
- c. When combining both approaches into the simulation scenario, the payload-RDL setting (P-RDL) performed the best for the parallel link arrangement.

Stage 5

In the last phase of this thesis, a novel simulation idea was constructed to combine the P-RDL scenario of the Quad-PILM TCP-based model with the use of adaptive

modulation (AM). Following what was done in Stage 4, this stage also implemented the segment switching and retry mechanisms into the Quad-PILM system.

The model was carefully designed during three steps of development to achieve better results. Gradual improvements of the goodput were shown, from the base AM scenario, and then followed by introducing the link value parameter analysis, and finally the updated version of the final arrangement.

In the final model, the results showed that the system executes the AM concept very well, as can be seen by the significant increases in goodput. In addition to the last proposed AM model, it was proven successful during the simulation under the extreme channel conditions tested. This again validates the AM model in the simulation.

7.2 Future Work

To end this thesis, some future work or improvements are suggested, particularly in the related fields discussed in this thesis. First, developing more functionality in the existing model would be useful. Some suggestions that can be made are:

- a. Following what has been discussed in Chapter 5, another set up can be made by implementing the system model by starting from lower or moderate QAM levels. By this, upward and downward adjustment could possibly happen and it would be interesting to analyze the effects of this on real channels. In fact this may be more relevant to actual deployed links that are likely to initialize at the lowest modulation level and then switch upwards in modulation if possible.
- b. As an improvement of the multi-link arrangement, to construct more links would be a challenge. A network device in future wireless networks implementing MIMO technology would probably have more than two (and maybe more than four) antennas. By improving this work model and implementing all the scenarios stated in this thesis, a recommendation could be made for such devices.

c. Imagine a system employing MIMO devices that is able to choose the best channel among the available channels, then to flexibly implement AM schemes. Flexible here means, for any particular time, the system should be able to switch to another better link given the new situation, whilst simultaneously making use of link adaptation techniques. In fact our system is able to be extended into this scenario - we already implement switching methods which can be applied extremely flexibly on a per-channel, per-packet basis. The main change needed would be to the switching heuristics themselves. This idea is comparable with what has been proposed in [127] called Adaptive MIMO.

Secondly, in terms of possible real world deployment, one would hope to see such a system installed and deployed in a monitoring scenario similar to Fig. 7.1. Initial investigations are very positive (see Appendix A), but more specific scenarios may need to be tested.

As discussed in this thesis, the models showed that significant goodput improvement was possible in the face of extreme channel differences – and this should match the conditions that may occur in the vicinity of a volcano: particularly regarding the unreliability of individual wireless channels due to problems such as node(s) displacement, path-loss, scattering, etc. Although this thesis has concentrated on MIMO

Figure 7.1: A typical multi-sensor volcano monitoring network deployment [1].

subchannels, the techniques developed here are equally applicable to any kind of parallel link – even those from different radios themselves, and potentially even links that travel over different physical media (which is a very possible scenario for volcano monitoring systems that often utilise several types of communications channels in order to increase robustness).

Appendices

Appendix A

Multi-point network

A dual- parallel link of multi-point network scenario is being simulated. The scenario model is depicted according Fig. A.1. Three different paths utilizing P-RDL method are examined, which are described as follows:

Path-1: $S - D$ (The original point-to-point model)

Path-2: $S - H_1 - D$

Path-3: $S - H_2 - H_3 - D$

For this setting, several assumptions are made:

- a. To maintain simplicity of our simulation network, dual-antenna connection between each hop is being used (i.e. two independent sub-channels per link).
- b. All the packets are transmitted on a "node-to-node" basis. This is due as the extension of our point-to-point model, which has its original purpose to maximize switching mechanism concepts between two nodes and to 'guarantee' the safety of packet arrival. This means, taking the example of Path-3: the whole packet from Node S must arrive safely at Node H_2 , prior to forwarding to Node H_3 and then onward until Node D - the last destination¹.

¹We notice that the scenario proposed here more likely will consume more power at each node. However, as the end user of this research work, i.e. the scientists, are more concerned with the arrival

Figure A.1: Multi-point network scenario.

- c. For each path, total channel delay would be made constant to maintain fairness (regardless how many hops are being used).

Path-2 observation

For the first setting, we compared Path-2 to the earliest (now called Path-1) model. As mentioned, for each hop taken for transmission, we maintained the total channel delay in the particular path similar to Path-1. To provide variations of our model, we simulated two scenarios that each hop has similar channel delay spread of $(1:1)$ and $(1:3)$, namely Link-2a and Link-2b respectively. For performance indicators, we consider goodput and overall transmission time for the test scenario data.

Refer to Fig. A.2(a) and Fig. A.2(b), for good BER environment of Path-2, when channel delay is spread similarly $(1:1)$, the goodput value shows the same tendency for each hop (Link-2a & Link-2b), though it is higher than the value of goodput in Path 1. For scenario $(1:3)$, the inclination is similar, but we found something interesting that when the channel delay is the least (in this case, a quarter of the original model

of the data, we considered that size might not be a big issue - solar panels and large batteries are being used. According to the scientists who have funded this work, with a real-world deployment costing several hundred thousand dollars, they prefer to spend money on extra batteries and solar panels if this will improve the availability of data. There is no concept of acceptable loss for their data - another reason why we chose to pursue the use of TCP/IP. Hence we do not follow the typical WSN trade-off as being primarily towards the lowest power in this particular instance.

(a)

(b)

(c)

(d)

Figure A.2: Goodput and Finishing time of Path-2, $\text{BER} = 10^{-5}$

Link-1), an improvement in the goodput can be noticed. From this, we argue that less delay in the channel seems to make the system allow more good bytes arrive safely. But note that this is for good BER conditions.

We considered another parameter, i.e. time taken to finish the transmission. Looking at Fig. A.2(c) and Fig. A.2(d), for each hop transmission (Link-2a and Link 2-b), each of the finishing times is very much alike also when compared to Path-1. However, this will add up to a total transmission time almost double of Path-1 only

Figure A.3: Goodput and Finishing time of Path-2, $\text{BER} = 10^{-4}$

(i.e. single hop transmission). This is obviously a trade-off that should be considered depending on the user requirements for timely data. And as for this research work, the lengthy time latency might not be a problem for the scientists, as long as they got the data in the end.

Just for comparison when considering worse BER conditions, as shown in Fig. A.3, we notice that the overall goodput value is decreased though the tendencies are similar for each hops goodput. It seems that for worse channel conditions, channel

delay does not necessarily impact the systems goodput. On the contrary, the overall transmission time increases significantly, specifically for the extreme channel differences. For each hop, it can be seen clearly that more time is needed to finish a transmission at each hop compared to Path-1. This is the contrary of the previous BER condition that almost has a similar timing. Here, it seems that making more hops over a bad channel will make the time to finish the transmission longer. Again, this is most likely due to the bad channel condition.

Path-3 observation

We also considered another set-up by employing another node in the Path-3. Similar scenarios and parameters are applied for this setting. However, since three links are used in this path, we assumed distribution value of the channel delay to be $(1:1:1)$ and $(1:2:3)$, with each link respectively named Link-3a, Link-3b, and Link-3c. The results for observation of Path-3 are given through Fig. A.4 and Fig. A.5.

For the first scenario when the channel delay for each link is distributed by $(1:1:1)$, there are slight noticeable improvements of goodput for each hop. Though not significant, this may be due to the possibility that for each hop transmission, most of the bad bits have been discarded and thus mainly only good data is sent. This also happens in worse BER channel, though the overall goodput is still higher at $\text{BER} = 10^{-5}$. Interestingly, when the channel delay is varied in the second scenario (i.e. $(1:2:3)$), the goodput shows the opposite behaviour. Higher goodput is gained when the particular channel experiences smaller delay. We argue that the less delay the packet has experienced, the more likely it will make the packet arrive safely at the destination. Note that both of these observations might be similar to Path-2.

In terms of finishing time, there is not much different for each hop. Major differences can be noticed when comparing the time taken to finish the transmission of Path-3 to Path-1, which also makes sense since Path-3 took much longer.

Figure A.4: Goodput and Finishing time of Path-3, $\text{BER} = 10^{-5}$

For worse channel conditions, the trend is similar with the exception that the overall goodput value is lower. For both channel delay scenarios, as can be guessed, for each delay increment the goodput decreases gradually. The observation of finishing time is also similar to the condition in good BER channel.

Figure A.5: Goodput and Finishing time of Path-3, $\text{BER} = 10^{-4}$

Remarks

Based on the performance of the simulated multi-point network, we conclude several points. In terms of goodput, adding more nodes can improve its value though the improvement is not really very significant. Quite the opposite may be true for a real deployment where it may introduce deployment and hardware cost. Moreover, though timing is not really an issue in the context of this research work, allowing multi-nodes would indeed yield a longer time to finish the transmission. The importance is

arguable though all depends on the application requirements.

Another closing remark: we found that there might be a tendency that similar channel delay distribution will make the bad data not be forwarded to the next hop. But taking in the overall sense of the system, reduced channel delay would yield an increment in goodput.

Lastly, our proposed model and results shows that the cross-layer techniques can indeed work in the multi-point scenario. Several trade-offs are there, but should be taken wisely according to the context of the work.

References

- [1] “Volcano Monitoring Sensor Network Architecture (Harvard Sensor Network Lab.).” [Online]. Available: <http://fiji.eecs.harvard.edu/Volcano>
- [2] K. Cho and D. Yoon, “On the general BER expression of one- and two-dimensional amplitude modulations,” *IEEE Trans. Commun.*, vol. 50, no. 7, pp. 1074–1080, Jul. 2002.
- [3] T. Rappaport, A. Annamalai, R. Buehrer, and W. Tranter, “Wireless communications: past events and a future perspective,” *IEEE Commun. Mag.*, vol. 40, no. 5, pp. 148–161, May 2002.
- [4] Y. Zhang and B.-H. Soong, “The effect of unreliable wireless channel on the call performance in mobile network,” *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 653–661, Mar. 2005.
- [5] T. S. Rappaport, *The Wireless communications: principles and practice*. NJ: Prentice-Hall, 1996.
- [6] A. Paulraj, D. Gore, R. Nabar, and H. Bolcskei, “An overview of MIMO communications - a key to gigabit wireless,” *Proc. IEEE*, vol. 92, no. 2, pp. 198–218, Feb. 2004.
- [7] “Earth Observatory of Singapore.” [Online]. Available: <http://www.earthobservatory.sg/index.html>

-
- [8] M. Ripepe, A. J. Harris, and R. Carniel, “Thermal, seismic and infrasonic evidences of variable degassing rates at stromboli volcano,” *Journal of Volcanology and Geothermal Research*, vol. 118, no. 34, pp. 285–297, 2002. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0377027302002986>
 - [9] P. J. Wallace, S. A. Cam, W. I. Rose, G. J. Bluth, and T. Gerlach, “Integrating petrologic and remote sensing perspectives on magmatic volatiles and volcanic degassing,” *Eos Trans. AGU*, vol. 84, no. 42, 2003.
 - [10] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, “Monitoring volcanic eruptions with a wireless sensor network,” in *Proc. 2nd European Workshop on Wireless Sensor Networks*, Jan. - 2 Feb. 2005, pp. 108–120.
 - [11] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, “Deploying a wireless sensor network on an active volcano,” *IEEE Internet Comput.*, vol. 10, no. 2, pp. 18–25, March-April 2006.
 - [12] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. LaHusen, “Air-dropped sensor network for real-time high-fidelity volcano monitoring,” in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys ’09. New York, NY, USA: ACM, 2009, pp. 305–318. [Online]. Available: <http://doi.acm.org/10.1145/1555816.1555847>
 - [13] W.-Z. Song, R. Huang, M. Xu, B. Shirazi, and R. LaHusen, “Design and deployment of sensor network for real-time high-fidelity volcano monitoring,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 11, pp. 1658 –1674, Nov. 2010.
 - [14] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, “Fidelity and yield in a volcano monitoring sensor network,” in *Proc. 7th symposium on Operating systems design and implementation*, ser. OSDI ’06. Berkeley, CA, USA: USENIX Association, 2006, pp. 381–396. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1298455.1298491>
-

-
- [15] I. McLoughlin and H. Sirisena, “TCP/IP link layer error mitigation for MIMO wireless links,” *J. Telecommun. Syst.*, 2010.
- [16] S. Prakash and I. McLoughlin, “Effects of channel prediction for transmit antenna selection with maximal-ratio combining in Rayleigh fading,” *IEEE Trans. Veh. Technol.*, vol. 60, no. 6, pp. 2555–2568, Jul. 2011.
- [17] “INET framework,” 2011. [Online]. Available: <http://inet.omnetpp.org/>
- [18] K. S. P. Richard M. Fujimoto and G. F. Riley, *Network Simulation*. Jean Walrand, University of California - Berkeley, 2007.
- [19] “CASTALIA - A Simulator for WSNs.” [Online]. Available: <http://castalia.npc.nicta.com.au/>
- [20] “MiXiM.” [Online]. Available: <http://mixim.sourceforge.net/>
- [21] “QualNet.” [Online]. Available: <http://www.scalable-networks.com/content/products/qualnet>
- [22] S. Floyd and V. Paxson, “Difficulties in simulating the internet,” *IEEE/ACM Trans. Netw.*, vol. 9, no. 4, pp. 392–403, Aug. 2001. [Online]. Available: <http://dx.doi.org.ezlibproxy1.ntu.edu.sg/10.1109/90.944338>
- [23] M. Wellens, M. Petrova, J. Riihijarvi, and P. Mahonen, “Building a better wireless mousetrap: need for more realism in simulations,” in *2nd Annu. Conf. on Wireless On-demand Network Systems and Services (WONS)*, Jan. 2005, pp. 150–157.
- [24] C. Singh, O. Vyas, and M. Tiwari, “A survey of simulation in sensor networks,” in *Proc. Int. Conf. Computational Intelligence for Modelling Control Automation*, Dec. 2008, pp. 867–872.

-
- [25] G. S. Fishman, *Principles of Discrete Event Simulation*. John Wiley & Sons, 1978.
 - [26] R. K. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons, 1992.
 - [27] P. Rathod, S. Perur, and R. Rangarajan, “Bridging the gap between reality and simulations: An ethernet case study,” in *9th Int. Conf. Information Technology*, Dec. 2006, pp. 52–55.
 - [28] “REAL Network Simulator.” [Online]. Available: <http://www.cs.cornell.edu/skeshav/real/overview.html>
 - [29] “ns-2.” [Online]. Available: <http://www.isi.edu/nsnam/ns/>
 - [30] T. R. Henderson, S. Roy, S. Floyd, and G. F. Riley, “ns-3 project goals,” in *Proc. Workshop on ns-2: the IP network simulator*, ser. WNS2 '06. New York, NY, USA: ACM, 2006. [Online]. Available: <http://doi.acm.org/10.1145/1190455.1190468>
 - [31] Y. Xue, H. S. Lee, M. Yang, P. Kumarawadu, H. Ghenniwa, and W. Shen, “Performance evaluation of NS-2 simulator for wireless sensor networks,” in *Canadian Conf. Electrical and Computer Engineering, 2007*, Apr. 2007, pp. 1372–1375.
 - [32] M. Takai, J. Martin, and R. Bagrodia, “Effects of wireless physical layer modeling in mobile ad hoc networks,” in *Proc. 2nd ACM Int. Symp. Mobile Ad Hoc Networking & Computing*, ser. MobiHoc '01. New York, NY, USA: ACM, 2001, pp. 87–94. [Online]. Available: <http://doi.acm.org.ezlibproxy1.ntu.edu.sg/10.1145/501426.501429>

-
- [33] K. M. Reineck, “Evaluation and comparison of network simulation tools,” Master’s thesis, University of Applied Sciences Bonn-Rhein-Sieg, 2008.
- [34] E. Weingartner, H. vom Lehn, and K. Wehrle, “A performance comparison of recent network simulators,” in *IEEE Int. Conf. Communications (ICC’09)*, Jun. 2009, pp. 1–5.
- [35] M. Ikeda, E. Kulla, L. Barolli, M. Takizawa, and R. Miho, “Performance evaluation of wireless mobile ad-hoc network via ns-3 simulator,” in *14th Int. Conf. Network-Based Information Systems (NBIS)*, Sep. 2011, pp. 135–141.
- [36] A. Bandyopadhyay, S. Vuppala, and P. Choudhury, “A simulation analysis of flooding attack in manet using ns-3,” in *2nd Int. Conf. Wireless Commun. Veh. Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*, Mar. 2011, pp. 1–5.
- [37] A. Paul, S. Konwar, U. Gogoi, A. Chakraborty, N. Yeshmin, and S. Nandi, “Implementation and performance evaluation of aodv in wireless mesh networks using ns-3,” in *Int. Conf. Education Technology and Computer (ICETC)*, vol. 5, Jun. 2010, pp. V5–298–V5–303.
- [38] X. Chang, “Network simulations with opnet,” in *Simulation Conference Proceedings, Winter*, vol. 1, 1999, pp. 307–314 vol.1.
- [39] J. Cai and B. Jia, “Network simulation based on OPNET and application,” in *1st Int. Workshop Education Technology and Computer Science, 2009*, vol. 1, Mar. 2009, pp. 199–202.
- [40] J. Dorleus, R. Holweck, Z. Ren, H. Li, H.-L. Cui, and J. Medina, “Modeling and simulation of fading and pathloss in OPNET for range communications,” in *IEEE Radio and Wireless Symp.*, Jan. 2007, pp. 407–410.

-
- [41] J. Hao, J. Wu, and C. Guo, “Modeling and simulation of can network based on opnet,” in *3rd Int. Conf. Communication Software and Networks (ICCSN), 2011*, May 2011, pp. 577–581.
 - [42] I. Hammoodi, B. Stewart, A. Kocian, and S. McMeekin, “A comprehensive performance study of OPNET modeler for ZigBee wireless sensor networks,” in *3rd Int. Conf. Next Generation Mobile Applications, Services and Technologies, 2009.(NGMAST '09)*, Sep. 2009, pp. 357–362.
 - [43] X. Xian, W. Shi, and H. Huang, “Comparison of OMNeT++ and other simulator for wsn simulation,” in *Proc. IEEE 3th. Industrial Electron. and Applications*, Jun. 2008, pp. 1439–1443.
 - [44] “OMNeT++.” [Online]. Available: <http://www.omnetpp.org/>
 - [45] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus, “Comparative study of wireless network simulators,” in *7th Int. Conf. Networking, 2008*, Apr. 2008, pp. 517–523.
 - [46] “Mobility Framework.” [Online]. Available: <http://mobility-fw.sourceforge.net/>
 - [47] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, “Simulating wireless and mobile networks in OMNeT++ the MiXiM vision,” in *Proc. 1st Int. Conf. Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools '08. ICST, Brussels, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 71:1–71:8. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1416222.1416302>
 - [48] Z. Zhang, Z. Pang, J. Chen, Q. Chen, H. Tenhunen, L.-R. Zheng, and X. Yan, “Two-layered wireless sensor networks for warehouses and supermarkets,”

- in *Proc. 2009 3rd Int. Conf. Conference Mobile Ubiquitous Computing, Systems, Services and Technologies*, ser. UBICOMM '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 220–224. [Online]. Available: <http://dx.doi.org/10.1109/UBICOMM.2009.16>
- [49] J.-C. Maureira, P. Uribe, O. Dalle, T. Asahi, and J. Amaya, “Component based approach using OMNeT++ for train communication modeling,” in *Int. Conf. ITS Telecommunications*, 2009.
- [50] X. Xian, W. Shi, and H. Huang, “Comparison of OMNeT++ and other simulator for WSN simulation,” in *3rd IEEE Conf. Industrial Electronics and Applications*, 2008, Jun. 2008, pp. 1439–1443.
- [51] D. Albeseder and M. Fuegger, “Small PC-network simulation - a comprehensive performance case study,” 2005, tech. Report.
- [52] E. Weingartner, H. vom Lehn, and K. Wehrle, “A performance comparison of recent network simulators,” in *Proc. IEEE Int. Conf. Communications*, 2009, Jun. 2009, pp. 1–5.
- [53] M. Gunther and K. Reineck, “Leistungsanalyse des olsr-protokolls in drahtlos vermaschten netzen mit ns-2,” 2008. [Online]. Available: <http://www.projektiv.net/karsten/Leistungsanalyse-des-OLSR-Protokolls-in-drahtlos-vermaschten-Netzen-mit-ns-2.pdf>
- [54] R. Han and D. Messerschmitt, “A progressively reliable transport protocol for interactive wireless multimedia,” *Multimedia Systems*, vol. 7, pp. 141–156, 1999, 10.1007/s005300050117. [Online]. Available: <http://dx.doi.org/10.1007/s005300050117>
- [55] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols (2nd Edition)*. Addison-Wesley Professional, 2011.

-
- [56] M. Thakur, H. Sirisena, and I. McLoughlin, “Link layer error mitigation in rural UHF-MIMO linking systems,” in *IEEE TENCON 2005 Region 10*, Nov. 2005, pp. 1–6.
- [57] B. Premkumar and C. Jianfei, *Principles of Wireless Communications and Networks (2nd Edition)*. Pearson Prentice Hall, 2006.
- [58] B. A. Forouzan, *TCP/IP Protocol Suite 2nd Ed.* McGraw-Hill, 2003.
- [59] M. Dulik, “TCP and UDP deployment in environment with high delay and high bit error rate,” in *Logistics and Industrial Informatics (LINDI), 2012 4th IEEE Symposium on*, Sep. 2012, pp. 131–134.
- [60] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach (6th Edition)*. Addison-Wesley, 2012.
- [61] I. Vijaya and A. Rath, “Simulation and performance evaluation of AODV, DSDV and DSR in TCP and UDP environment,” in *3rd Int. Conf. Electronics Computer Technology (ICECT), 2011*, vol. 6, Apr. 2011, pp. 42–47.
- [62] J. Kamal, M. Hasan, A. Carrington, and H. Yu, “Lessons learned from real MANET experiments and simulation-based evaluation of UDP and TCP,” in *13th Int. Conf. Computer and Information Technology (ICCIT), 2010*, Dec. 2010, pp. 511–515.
- [63] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla, “The impact of multihop wireless channel on TCP performance,” *IEEE Trans. Mobile Computing*, vol. 4, no. 2, pp. 209–221, Apr. 2005.
- [64] M. Gerla, K. Tang, and R. Bagrodia, “TCP performance in wireless multihop networks,” in *2nd IEEE Proc. Workshop Mobile Computing Systems and Applications*, Feb. 1999, pp. 41–50.

-
- [65] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and A. Sivakumar, "ATP: a reliable transport protocol for ad hoc networks," *IEEE Trans. Mobile Computing*, vol. 4, no. 6, pp. 588–603, Dec. 2005.
- [66] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 756–769, Dec. 1997.
- [67] H. Karl and A. Willig, *Protocols and Architecture for Wireless Sensor Networks*. John Wiley & Sons, 2005.
- [68] C. Wang, K. Sohraby, B. Li, M. Daneshmand, and Y. Hu, "A survey of transport protocols for wireless sensor networks," *IEEE Network*, vol. 20, no. 3, pp. 34–40, may-june 2006.
- [69] A. Bakre and B. Badrinath, "I-TCP: indirect TCP for mobile hosts," in *Proc. 15th Int. Conf. Distributed Computing Syst.*, Jun. 1995, pp. 136–143.
- [70] G. Xylomenos, G. Polyzos, P. Mahonen, and M. Saaranen, "TCP performance issues over wireless links," *IEEE Commun. Mag.*, vol. 39, no. 4, pp. 52–58, Apr. 2001.
- [71] K. Chandran, S. Ragbunathan, S. Venkatesan, and R. Prakash, "A feedback based scheme for improving TCP performance in ad-hoc wireless networks," in *Proc. 18th Int. Conf. Distributed Computing Syst.*, May 1998, pp. 472–479.
- [72] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad-hoc networks," *Wireless Network*, vol. 8, no. 2/3, p. 275288, Mar. 2002. [Online]. Available: <http://dx.doi.org.ezlibproxy1.ntu.edu.sg/10.1023/A:1013798127590>
- [73] X. Chen, H. Zhai, J. Wang, and Y. Fang, "TCP performance over mobile ad hoc networks," *Electrical and Computer Engineering, Canadian Journal of*, vol. 29, no. 1/2, pp. 129–134, january/april 2004.
-

-
- [74] J. Liu and S. Singh, “ATCP: TCP for mobile ad hoc networks,” *IEEE J. Select. Areas Commun.*, vol. 19, no. 7, pp. 1300–1315, Jul. 2001.
 - [75] P. S.-J. S. K. Anantharaman, V. and R. Sivakumar, “TCP performance over mobile ad hoc networks: a quantitative study,” *J. Wireless Commun. & Mobile Computing*, vol. 4, pp. 203–222, Mar. 2004.
 - [76] K. Xu, M. Gerla, L. Qi, and Y. Shu, “Enhancing TCP fairness in ad hoc wireless networks using neighborhood RED,” in *Proc. 9th Annu. Int. Conf. Mobile Computing and Networking*, ser. MobiCom ’03. New York, NY, USA: ACM, 2003, pp. 16–28. [Online]. Available: <http://doi.acm.org/10.1145/938985.938988>
 - [77] S. Shakkottai, T. Rappaport, and P. Karlsson, “Cross-layer design for wireless networks,” *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 74–80, Oct. 2003.
 - [78] V. Kawadia and P. Kumar, “A cautionary perspective on cross-layer design,” *IEEE Wireless Commun.*, vol. 12, no. 1, pp. 3–11, Feb. 2005.
 - [79] E. Anggadajaja and I. McLoughlin, “Point-to-point OMNeT++ based simulation of reliable transmission using realistic segmentation and reassembly with error control,” in *2nd Int. Conf. Advances in Computing, Control and Telecommunication Technologies (ACT)*, Dec. 2010, pp. 125–128.
 - [80] A. Naguib, N. Seshadri, and A. Calderbank, “Increasing data rate over wireless channels,” *IEEE Signal Processing Mag.*, vol. 17, no. 3, pp. 76–92, may 2000.
 - [81] H. Jafarkhani, *Space-Time Coding: Theory and Practice*. Cambridge University Press, 2005.
 - [82] A. Chandra, *Diversity Combining for Digital Signals in Wireless Fading Channels: Analysis and Simulation of Error Performance*. LAP LAMBERT Academic Publishing, 2011.
-

-
- [83] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Sel. Areas Commun.*, pp. 1451–1458, 1998.
 - [84] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, 1996. [Online]. Available: <http://dx.doi.org/10.1002/bltj.2015>
 - [85] D. Gesbert, M. Shafi, D. shan Shiu, P. Smith, and A. Naguib, "From theory to practice: An overview of MIMO space-time coded wireless systems," *IEEE J. Sel. Areas Commun*, vol. 21, no. 3, pp. 281–302, Apr. 2003.
 - [86] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Commun.*, vol. 6, pp. 311–335, 1998.
 - [87] E. Telatar, "Capacity of multi-antenna gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, 1999. [Online]. Available: <http://dx.doi.org/10.1002/ett.4460100604>
 - [88] A. Paulraj and C. Papadias, "Space-time processing for wireless communications," *IEEE Signal Processing Mag.*, vol. 14, no. 6, pp. 49–83, Nov. 1997.
 - [89] D. Gesbert and J. Akhtar, "Breaking the barriers of shannon's capacity: An overview of MIMO wireless systems," 2002.
 - [90] A. Toledo and X. Wang, "TCP performance over wireless MIMO channels with ARQ and packet combining," *IEEE Trans. Mobile Computing*, vol. 5, no. 3, pp. 208–223, Mar. 2006.
 - [91] M. Park, S.-H. Choi, and S. Nettles, "Cross-layer MAC design for wireless networks using MIMO," in *Proc. IEEE Global Telecommunications Conf. (GLOBE-COM '05)*, vol. 5, Dec. 2005.
-

-
- [92] Z. Zhou, B. Vucetic, M. Dohler, and Y. Li, "MIMO systems with adaptive modulation," *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1828–1842, Sep. 2005.
 - [93] Y. Lin and V. Wong, "Cross-layer design of MIMO-enabled WLANs with network utility maximization," *IEEE Trans. Veh. Technol.*, vol. 58, no. 5, pp. 2443–2456, Jun. 2009.
 - [94] X.-B. Yu, S.-H. Leung, Q. Kuang, and X. Yin, "Performance analysis of variable-power adaptive modulation with outdated feedback for space-time coded MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2613–2624, Jul. 2012.
 - [95] J. Park and S. Lee, "Distributed MIMO ad-hoc networks: Link scheduling, power allocation, and cooperative beamforming," *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2586–2598, Jul. 2012.
 - [96] K. Mehrotra and I. McLoughlin, "Time reversal space time block coding with channel estimation errors," in *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, vol. 1, 2003, pp. 617–620 Vol.1.
 - [97] A. Camargo, *Adaptive Modulation, Channel Coding and MIMO Schemes for Practical OFDM Systems*.
 - [98] A. Goldsmith and P. Varaiya, "Capacity of fading channels with channel side information," *IEE Tran. Inf. Theory*, vol. 43, no. 6, pp. 1986–1992, Nov. 1997.
 - [99] J. Hayes, "Adaptive feedback communications," *IEEE Trans. Commun. Technol.*, vol. 16, no. 1, pp. 29–34, Feb. 1968.
 - [100] J. Cavers, "Variable-rate transmission for rayleigh fading channels," *IEEE Trans. Commun.*, vol. 20, no. 1, pp. 15–22, Feb. 1972.
-

-
- [101] G. Pottie, "System design choices in personal communications," *IEEE Personal Commun.*, vol. 2, no. 5, pp. 50–67, Oct. 1995.
- [102] W. Webb and R. Steele, "Variable rate QAM for mobile radio," *IEEE Trans. Commun.*, vol. 43, no. 7, pp. 2223–2230, Jul. 1995.
- [103] M.-S. Alouini and A. Goldsmith, "Capacity of Nakagami multipath fading channels," in *IEEE 47th Vehicular Technology Conf., 1997*, vol. 1, May 1997, pp. 358–362.
- [104] —, "Capacity of rayleigh fading channels under different adaptive transmission and diversity-combining techniques," *IEEE Trans. Veh. Technol.*, vol. 48, no. 4, pp. 1165–1181, Jul. 1999.
- [105] M.-S. Alouini and A. J. Goldsmith, "Adaptive modulation over Nakagami fading channels," *Wireless Personal Commun.*, vol. 13, pp. 119–143, 2000.
- [106] A. Goldsmith and S.-G. Chua, "Variable-rate variable-power MQAM for fading channels," *IEEE Trans. Commun.*, vol. 45, no. 10, pp. 1218–1230, Oct. 1997.
- [107] —, "Adaptive coded modulation for fading channels," *IEEE Trans. Commun.*, vol. 46, no. 5, pp. 595–602, May 1998.
- [108] K. Hole, H. Holm, and G. Oien, "Adaptive multidimensional coded modulation over flat fading channels," *IEEE J. Select. Areas Commun.*, vol. 18, no. 7, pp. 1153–1158, Jul. 2000.
- [109] M. Pursley and J. Shea, "Adaptive nonuniform phase-shift-key modulation for multimedia traffic in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 18, no. 8, pp. 1394–1407, Aug. 2000.
- [110] S. T. Chung and A. Goldsmith, "Degrees of freedom in adaptive modulation: a unified view," *IEEE Trans. Commun.*, vol. 49, no. 9, pp. 1561–1571, Sep. 2001.
-

-
- [111] B. Choi and L. Hanzo, "Optimum mode-switching-assisted constant-power single- and multicarrier adaptive modulation," *IEEE Trans. Veh. Technol.*, vol. 52, no. 3, pp. 536–560, May 2003.
 - [112] L. T. Ong, M. Shikh-Bahaei, and J. Chambers, "Variable rate and variable power MQAM system based on bayesian bit error rate and channel estimation techniques," *IEEE Trans. Commun.*, vol. 56, no. 2, pp. 177–182, Feb. 2008.
 - [113] D. Goeckel, "Adaptive coding for time-varying channels using outdated fading estimates," *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 844–855, Jun. 1999.
 - [114] S. Prakash and I. McLoughlin, "Analysis of adaptive modulation with antenna selection under channel prediction errors," in *Proc. Int. Conf. Comms. and Signal Process.*, Calicut, India, Feb. 2011.
 - [115] J. Huang and S. Signell, "On performance of adaptive modulation in MIMO systems using orthogonal space time block codes," *IEEE Trans. Veh. Technol.*, vol. 58, no. 8, pp. 4238–4247, Oct. 2009.
 - [116] Q. Liu, S. Zhou, and G. Giannakis, "Queuing with adaptive modulation and coding over wireless links: cross-layer analysis and design," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 1142–1153, May 2005.
 - [117] J. Harsini and F. Lahouti, "Queuing with adaptive modulation over MIMO wireless links for deadline constrained traffic: Cross-layer analysis and design," in *Proc IEEE Int. Conf. Communications (ICC '07)*, Jun. 2007, pp. 235–240.
 - [118] Q. Liu, S. Zhou, and G. Giannakis, "Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1746–1755, 2004.
 - [119] C. Gonzalez, L. Szczecinski, and S. Aissa, "Throughput maximization of ARQ transmission protocol employing adaptive modulation and coding," in *Telecom-*
-

- munications and Networking - ICT 2004*, ser. Lecture Notes in Computer Science, J. de Souza, P. Dini, and P. Lorenz, Eds. Springer Berlin / Heidelberg, 2004, vol. 3124, pp. 33–44.
- [120] C. Caini, R. Firrincieli, and D. Lacamera, “The TCP ”adaptive-selection” concept,” *IEEE Syst. J.*, vol. 2, no. 1, pp. 83–89, Mar. 2008.
 - [121] L. Le, E. Hossain, and T. Le-Ngoc, “Interaction between radio link level truncated ARQ, and TCP in multi-rate wireless networks: A cross-layer performance analysis,” *IET Commun.*, vol. 1, no. 5, pp. 821–830, Oct. 2007.
 - [122] Q. Liu, S. Zhou, and G. Giannakis, “TCP performance in wireless access with adaptive modulation and coding,” in *Proc. IEEE Int. Conf. Communications*, vol. 7, Jun. 2004, pp. 3989–3993.
 - [123] L. Zou, Y. Zhao, B. Wang, and Q. Liang, “The effects of adaptive modulation on the TCP performance,” in *IEEE 2002 Int. Conf. Communications, Circuits and Systems and West Sino Expositions*, vol. 1, June-1 July 2002, pp. 262–266.
 - [124] H. Rath, A. Karandikar, and V. Sharma, “Adaptive modulation-based TCP-aware uplink scheduling in IEEE 802.16 networks,” in *IEEE Int. Conf. Communications, 2008*, May 2008, pp. 3230–3236.
 - [125] A. Selvam, K. Kathiravan, and R. Reshmi, “A cross-layer TCP protocol with adaptive modulation for MANETs,” in *Int. Conf. Signal Processing, Communications and Networking, 2008*, 2008, pp. 428–433.
 - [126] E. Anggadaja and I. V. McLoughlin, “Modelling WSNs using OMNeT++,” in *Technological Advancements and Applications in Mobile Ad-Hoc Networks: Research Trends*, K. I. Lakhtaria, Ed. IGI Global, 2012.
 - [127] C.-B. Chae, A. Forenza, R. Heath, M. McKay, and I. Collings, “Adaptive MIMO transmission techniques for broadband wireless communication systems [topics

in wireless communications],” *IEEE Commun. Mag.*, vol. 48, no. 5, pp. 112–118, May 2010.