Transport Layer Performance Enhancements over Wireless Networks

Toktam Mahmoodi

Centre for Telecommunications Research

Department of Electronic Engineering
King's College London

A thesis submitted to the University of London
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
August 2009

Abstract

We are currently witnessing a rapidly increasing number of mobile users utilising the Transmission Control Protocol (TCP) over wireless networks for accessing services over the Internet. TCP has been designed for wireline networks and its shortcomings over wireless networks, such as throughput degradation due to random losses and intermittent connectivity, have been the subject of a large volume of research investigations over the last few years. In this thesis, a set of techniques are proposed to enhance the performance of the end-to-end wireless communications using TCP as the transport layer protocol. The proposed set of techniques use information from the TCP connections such as Round Trip Time (RTT), congestion window (cwnd), and the flavour of TCP. In that respect, the wireless link-layer algorithms are highly adaptive to these parameters that steer the performance of TCP. In the design of such a smart link-layer, important issues such as complexity, scalability and stability are being considered. Through an extensive set of simulations the performance of the proposed techniques is investigated thoroughly, with the focus on the figures of merits that affect the experience of the end-user. In this respect, the end-to-end throughput that a TCP flow can accomplish and the fairness among end-to-end flows belonging to possibly different flavours of TCP are examined.

Acknowledgements

The journey to my PhD has been long enough to enjoy, and short enough to always run out of time. It had its fair share of ups and downs. The happier times were as pleasing as one could imagine; the not-so-happy ones sometimes brought me to the brink of insanity. Nonetheless, it was an experience that taught me much more years worth of knowledge. First and foremost, I am deeply indebted to my supervisor Dr. Vasilis Friderikos, who supported me in this journey. He taught me not only how to conduct high quality research but also to keep up certain moral values as a researcher. I will always be thankful to him and respect him immensely for teaching me the different facets of a good researcher.

I would also like to thank Prof. Hamid Aghvami for giving me the opportunity to join CTR as a PhD student. The effort he constantly puts to provide the pleasant research environment at CTR, and the high standards he sets for research is always a great motivation for all the researchers. He was greatly supportive and motivating during my PhD.

I want to thank all industrial and academic members of Mobile VCE Core 4, Efficiency research program for their support. Researching alongside Mobile VCE members was a truly enlightening experience and taught me a great deal from both the academic and industrial contexts of research. Those Mobile VCE meetings were among the very different experiences I could have achieved during my PhD.

CTR is probably one of the friendliest places to work. I have enjoyed the company of many of my colleagues and could make lots of valuable friends. Raquel with no doubt was the first and the best friend I made in

CTR. Although she was in CTR for such a short period, we challenged and explored many many experiences together from living in London to working with OPNET. It was all a great fun alongside the work that made my life in Stanford street so comfortable and pleasant. Oliver, whom without his support some of this work would not have been possible. He supported me alongside my research, and with his punctuality and perfectionism taught me a lot. I still hope I can do things with small percentage of his precision. Room 113 with Imran and Kostas was the best break for me whenever I stocked in my work for any reason. Kostas and I may have not come to a conclusion that which one of the Persian or Greek empire have been greater, but I certainly enjoyed those historical discussions. Having a chat with these two would have been enough to send me back to work! I should also mention that Imran has kindly provided me very constructive feedbacks over my thesis that I am particularly grateful for them. Paul was probably the most welcoming face I encountered in CTR right after I joined in 2006. He is always there to help and support in any means. Although most of knocking at door 112 was for Paul, I have always enjoyed Andrej being there and those few short and long technical, psychological, political, religious discussions we had together. Nima with his adventurous character is a great friend I had the opportunity to meet in CTR. His passion for music, and his similar taste of music to me was a good enough reason for a pleasant company in going to the concerts. Unfortunately, we never had a chance to play music together which I hope we can make it later. Dev and George who I have spent most of my lunch time with, are among the very good friend I have in CTR. We started almost at the same time and went through ups and downs of this journey together. I enjoyed their company and our either study related or non-related discussions. Thikriat and Nika were always the feminine face of CTR for me with whom you do not need to pretend being somebody else stronger than yourself; you are accepted as you are. Room 111 with these two girls if not the best, is among the

sweetest places one can find in CTR. Alireza was the first who welcomed me in the network room, and the so many over lunch chat we had, made those moments unforgettable. I am grateful of Haffiz's company and wish him all the best in his PhD; he is a friend who can always put a smile on my face even in the toughest of time. For the rest of my colleagues at CTR, my thanks (in no particular order) go to Uma, Stan, Panagiotis, Sampath, Dev Senior, Mojdeh, Merat, Fernando, Laura, Aimal, Nancy, Vivien, Reza, Naveel, Mona, Dr. Reza Nakhai, and Dr. Fatin Said, for providing such a great company.

Outside CTR, I wish to thank all my friends who supported me whenever needed. Yassaman who is my forever schoolmate and welcomed me in Heathrow for the very first time. Khatereh whom her passionate character made the year she was in London a memorable year for me. Bahman, an old friend whom with his neighborhood office and his delicate character was always the best mate for a chat over a cup of tea. Hasti, Parastoo, and Leyla filled the place of my lovely friends from back home and let me feel London home. I am particularly thankful to Behrang and Payman, new members ofKing's math department, who have reviewed my thesis and have provided me valuable comments.

I also wish to express my deep gratitude to my friends back home, who were my company in running the race of life and had a great effect on my decision to pursue my education and join the PhD programme. I laughed and cried with them in the pleasure and sadness of life, and will do in the future. I have a very big thank to my mountaineering group back home, Gonzales, for making my summer holidays the best possible with giving me an opportunity to enjoy my favourite activity with the best mates ever.

My last but not least thoughts go to my parents. I cannot find words to do justice to my Mom and Dad. My parents have always provided unconditional support and appreciation for my scholarly endeavors. They

taught me how to aim high in life and succeed with true devotion. With no doubt I can say they are the only ones who understand me and love me in all circumstances. My sisters, Arezoo and Haniye, whom I love them among the most, and whom their appreciation have always been the very strong reasons for me to experience and explore new things in life. I would have loved to be a good role model as their older sister and I hope I could.
I dedicate my thesis to all my beloved ones, either named here or not, in my 30 years of life.

Toktam

Contents

Abbreviations **12**

List of Figures **16**

List of Tables **19**

1 Introduction **20**

1.1 Scope of the Work 21

1.2 Contributions of the Thesis 22

1.3 Structure of the Thesis 24

2 Background Study **26**

2.1 Transport Layer 28

2.1.1 Transmission Control Protocol (TCP) 28

2.1.1.1 TCP Connection Establishment and Termination . . 28

2.1.1.2 TCP Timer Management 29

2.1.1.3 TCP Congestion Control Algorithm 30

2.1.1.4 TCP Enhanced Congestion Control 31

2.1.2 TCP Variations and Flavours 32

2.1.2.1 Baseline TCP's Congestion Control 32

2.1.2.2 TCP Over Dynamic Traffic Loads 34

2.1.2.3 TCP Over High Speed Links 36

2.1.2.4 TCP over Asymmetric Paths 37

2.1.3 Other Transport Protocols 38

2.1.3.1 User Datagram Protocol (UDP) 38

2.1.3.2 TCP-Friendly Rate Control (TFRC) 38
2.1.3.3 Datagram Congestion Control Protocol (DCCP) . . 39
2.1.4 TCP Throughput Modelling 39
2.1.4.1 Modelling TCP Reno 40
2.1.4.2 Modelling Enabled SACK option 40
2.1.4.3 Modelling TCP NewReno 41
2.1.4.4 Modelling TCP Westwood 41
2.2 Evolution of Wireless Networks 42
2.2.1 Wireless Local Area Network (WLAN) 43
2.2.2 Worldwide inter-operability for Microwave Access (WiMAX) 44
2.2.3 Long Term Evolution (LTE) 44
2.2.4 Long Term Evolution (LTE)-Advanced 45
2.3 TCP over Wireless and Mobile Networks 45
2.3.1 End-to-End Solutions 46
2.3.2 Split TCP Connection 47
2.3.3 Link-Level Solutions 48
2.4 Cross-Layer Approaches towards TCP 49
2.4.1 TCP and Automatic Repeat reQuest (ARQ) 50
2.4.2 TCP and Forward Error Correction (FEC) 52
2.4.3 TCP and Resource Allocation 54
2.4.4 Cross-Layer Interactions between TCP and Link-Layer 55

3 TCP-aware Link-Level FEC to Improve End-to-End Performance 57
3.1 Background Study 59
3.1.1 FEC Adaptation to Improve TCP Performance 60
3.1.2 Fairness Among TCP Flows in Wireless Networks 61
3.1.3 Measure of Fairness: Jain's Fairness Index 61
3.2 Throughput Comparison of Various TCP Flavours 62
3.3 Fairness among TCP flows in Multiple-Flavour Scenarios 63
3.3.1 Problem Definition 64

3.3.2 Link-Layer FEC Code Rate Selection 65
3.4 Observations Based on Numerical Investigations 66
3.4.1 Numerical Results 67
3.4.2 Using the Logarithmic Barrier Method to Solve the Problem in Real-Time 69
3.5 Performance Investigations 71
3.5.1 Simulation Parameters 71
3.5.2 Convolutional Encoder/Decoder 72
3.5.3 Simulation Scenarios and Simulation Results 74
3.5.3.1 Simulation Scenario One: Various end-to-end RTT values 75
3.5.3.2 Simulation Scenario Two: Changing the users' distance from the AP 76
3.5.3.3 Simulation Scenario Three: Different traffic types . . 77
3.5.3.4 Overall Results 78
3.6 Concluding Remarks 79

4 TCP-aware ARQ Mechanism to Improve End-to-End Performance 80
4.1 Background Study 82
4.2 TCP-Aware Dynamic ARQ Mechanism 83
4.2.1 Algorithm Description 84
4.3 Queuing Analysis of the TCP-Aware Dynamic ARQ Mechanism . . . 86
4.4 Performance Investigations 89
4.4.1 Simulation Parameters 89
4.4.2 Simulation Scenarios and Simulation Results 90
4.4.2.1 Simulation Scenario One: Changing the users' distance from the AP 91
4.4.2.2 Simulation Scenario Two: Different distribution of the end-to-end RTTs 91
4.4.2.3 Simulation Scenario Three: Various number of mobile users 92

4.4.2.4 Simulation Scenario Four: Random distribution of users . . . 92
4.4.2.5 Overall Results . . . 93
4.5 Concluding Remarks . . . 94

5 TCP-Aware Resource Allocation Schemes 95
5.1 Background Study . . . 98
5.1.1 OFDMA Resource Allocation Schemes . . . 98
5.1.2 TCP-Aware Resource Allocation Schemes . . . 99
5.1.3 TCP over Asymmetric Links . . . 100
5.2 System Model . . . 100
5.3 Downlink Resource Allocation Problem . . . 102
5.3.1 Proportional TCP Throughput Constrained (P1) . . . 102
5.3.2 Rate Difference from TCP Throughput Constrained (P2) . . . 103
5.4 Joint Uplink-Downlink Resource Allocation Problem . . . 104
5.4.1 Uplink Requirements . . . 105
5.4.2 Proportional TCP Throughput Constraint (P3) . . . 106
5.4.3 Rate Difference from TCP Throughput Constraint (P4) . . . 107
5.5 Optimal Subcarrier Allocation and Power Distribution . . . 108
5.6 Suboptimal Subcarrier Allocation and Power Distribution . . . 109
5.6.1 Downlink Resource Allocation Scheme (P1) . . . 110
5.6.2 Downlink Resource Allocation Scheme (P2) . . . 112
5.6.3 Joint Uplink-Downlink Resource Allocation Scheme (P3) . . . 113
5.6.4 Joint Uplink-Downlink Resource Allocation Scheme (P4) . . . 114
5.7 Performance Investigations . . . 116
5.7.1 Fairness Analysis . . . 116
5.7.2 Simulation Parameters . . . 116
5.7.3 Downlink Resource Allocation Schemes (P1 and P3) . . . 118
5.7.3.1 Primary observations on the resource allocation (P1) 118
5.7.3.2 Investigations on the selection of μ for the resource allocation scheme (P2) . . . 119

5.7.3.3 Investigations on the performance of (P1 and P2) . . 121
5.7.4 Joint Uplink-Downlink Resource Allocation Schemes (P3 and P4) 123
5.7.4.1 Investigations on the performance of (P3 and P4) . . 124
5.8 Concluding Remarks 125

6 Conclusions and Future Research 130
6.1 Concluding Remarks 130
6.2 Avenues of Future Research 133

References 145

A Details and Proofs of the used Mathematical Models 146
A.1 Newton's Method in Solving Optimisation Problems 146
A.1.1 The Newton step 147
A.1.2 Newton's method 148
A.2 Concept of the Logarithmic Barrier Function 148
A.2.1 Example Problem 150

B Simulation Modelling 151
B.1 OPNET Modeler 151
B.1.1 Why OPNET Modeler 151
B.1.2 Modelling Methodology 152
B.1.2.1 Network Editor 152
B.1.2.2 Node Editor 153
B.1.2.3 Process Editor 154
B.1.3 Wireless Modelling 155
B.1.3.1 Wireless Channels 155
B.1.3.2 Modifications to the Standard Library 157
B.1.4 Transport Layer Modelling 157
B.1.4.1 Modifications to the Standard Library: TCP Westwood 158
B.2 Network Configuration 158

Abbreviations

2G 2nd Generation

3G 3rd Generation

3GPP 3rd Generations Partnership Projects

ACK ACKnowledge

AIMD Additive Increase, Multiplicative Decrease

AP Access Point

API Application Programming Interface

ARQ Automatic Repeat reQuest

BER Bit Error Rate

BPSK Binary Phase Shift Keying

CDMA Code Division Multiple Access

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CTCP Compound TCP

cwnd congestion window

DCCP Datagram Congestion Control Protocol

DES Discrete Event Simulation

DupACK Duplicate ACK

E-UTRA Evolved UMTS Terrestrial Radio Access

ECN Explicit Congestion Notification

ERE Eligible Rate Estimation

FDMA Frequency Division Multiple Access

FEC Forward Error Correction

FIN Finish Flag

FSM Finite State Machine

FTP File Transfer Protocol

GPRS General Packet Radio Service

GSM Global System for Mobile communications

HARQ Hybrid ARQ

HSDPA High-Speed Downlink Packet Access

HSTCP High Speed TCP

HTTP Hypertext Transfer Protocol

I-TCP Indirect-TCP

ICI Interface Control Information

IP Internet Protocol

IMT International Mobile Telecommunications

ITU International Telecommunication Union

LLE-TCP Link-Layer ARQ Exploitation TCP

LTE Long Term Evolution

M-QAM M-array Quadrature Amplitude Modulation

M-TCP Mobile TCP

MAC Medium Access Control

MIMD Multiplicative Increase, Multiplicative Decrease

MIMO Multiple Input, Multiple Output

MSS Maximum Segment Size

NACK Negative ACK

OFDM Orthogonal Frequency Division Multiplexing

OFDMA Orthogonal Frequency Division Multiple Access

P-K Pollacszek-Khinchin

PER Packet Error Rate

PHY Physical Layer

PL Path Loss

QAM Quadrature Amplitude Modulation

QoS Quality of Service

QPSK Quadrature Phase Shift Keying

RFC Request For Comment

RTO Retransmission Time Out

RTT Round Trip Time

SACK Selective Acknowledgement

SC-FDMA Single Carrier-FDMA

SNR Signal to Noise Ratio

ssthresh slow start threshold

STCP Scalable TCP

SYN Synchronize sequence numbers

TACS Total Access Communications System

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

TFRC TCP-Friendly Rate Control

UDP User Datagram Protocol

UMTS Universal Mobile Telecommunications System

UTRA UMTS Terrestrial Radio Access

VoIP Voice over IP

WiMAX Worldwide inter-operability for Microwave Access

WLAN Wireless Local Area Network

List of Figures

2.1 The structure of Chapter 2. 27
2.2 Cross-Layer Interactions between TCP at the end-host and the Link-Layer at the wireless base station. 55

3.1 Single-flow analytical model and simulation results comparisons for TCP Reno, TCP Reno+SACK, TCP NewReno, and TCP Westwood 63
3.2 The investigated network architecture. 67
3.3 Numerical observations from the TCP-aware FEC scheme compared with what would otherwise be achieved in scenario 3. 68
3.4 CDF of the number of iterations required in solving the optimisation problem using Newton's method 71
3.5 TCP packet loss versus link BER in various Convolutional code rates with K = 7, $g_0 = 171$ and $g_1 = 133$. 74
3.6 Users' distribution around the wireless AP while keeping fixed distances from the AP. 75
3.7 Scenario one: Jain's index versus end-to-end RTT (ms). 76
3.8 Scenario two: Jain's index versus users' distance from the AP (m). . . 77
3.9 Scenario two: Overall end-to-end throughput versus users' distance from the AP (m) 78

4.1 RTT over the wireless link is by far smaller than the end-to-end RTT. 82
4.2 Scenario one: Aggregated end-to-end throughput versus wireless users' distance from the AP. 91

4.3 Scenario three: End-to-End aggregated throughput versus number of mobile users while users' distance from the AP is 55 m 93

5.1 Downlink and uplink capacity of the wireless link 105
5.2 Effect of link asymmetry on the TCP throughput: Downlink throughput versus link BER, downlink capacity is 20 Mbps 106
5.3 The flow chart that corresponds to Algorithm 5. 115
5.4 Fairness Index as achieved by solving resource allocation problems (P1), and (BM2) in two-user scenario versus the proportion of two TCP flows' theoretical throughput. 119
5.5 Two-User Scenario (Problem P2): Total differences between the average TCP throughput and the actual data rate on wireless link as achieved by resource allocation problem (P2) versus μ, shown on the red (light grey) bar; utilised capacity of wireless link as achieved by resource allocation problem (P2) versus μ, shown on the blue (dark grey) bar. 120
5.6 Ten-User Scenario (Problem P2): Total differences between the average TCP throughput and actual data rate on wireless link as achieved by resource allocation problem (P2) versus μ, shown on the red (light grey) bar; utilised capacity of wireless link as achieved by resource allocation problem (P2) versus μ, shown on the blue (dark grey) bar. 121
5.7 Ten-User Scenario(Problem P2): Mean Value of fairness index as achieved by resource allocation problem (P2) versus μ, shown on the left axis; utilised capacity of wireless link as achieved by resource allocation problem (P2) versus μ, shown on the right axis. 122
5.8 Fairness Index as achieved by solving resource allocation problems (P1), (P2), (BM1), and and (BM2) versus the number of mobile users.123
5.9 Sum data rate (Mbps) as achieved by solving resource allocation problems (P1), (P2), (BM1), and and (BM2) versus the number of mobile users. 124

5.10 Fairness Index as achieved by solving resource allocation problems (P3), (P4), (BM1), and (BM2) versus the number of mobile users. . . 125
5.11 Downlink sum data rate (Mbps) as achieved by solving resource allocation problems (P3), (P4), (BM1), and (BM2) versus the number of mobile users. 126
5.12 Total (Uplink+Downlink) sum data rate (Mbps) as achieved by solving resource allocation problems (P3), (P4), (BM1), and (BM2) versus the number of mobile users. 126

B.1 Hierarchy of the OPNET modelling 159
B.2 Wireless Link Pipeline Stages 160

List of Tables

3.1 Fairness improvement as achieved by the proposed TCP-aware FEC scheme. 68
3.2 Simulation Parameters for the TCP-aware FEC scheme 73
3.3 Jain's Index in scenarios three using FTP and HTTP applications . . 79

4.1 Simulation Parameters for the TCP-aware ARQ scheme 90
4.2 Scenario two: End-to-End aggregated throughput for the Uniform, Normal and Exponentially distributed RTTs 92
4.3 Average percentage of TCP retransmitted packets using the TCP-aware ARQ mechanism. 94

5.1 Simulation Parameters for the TCP-aware Resource Allocation Scheme. 117

Chapter 1

Introduction

Internet applications have become part of everyday life within the last decades. In addition to the applications such as E-mail, Instant Messaging, others such as video streaming, Voice over IP (VoIP), and peer-to-peer are in widespread use today. Portable laptop computers along with their embedded wireless devices which were the replacement for traditional desktop computers accessing the Internet, are altered with the mobile phones. Hence, recent mobile phones with their advanced network capabilities and strong processing capacity are expected to be used for a variety of Internet applications as extensively as the fixed desktops or portable laptops are used.

The end-to-end design philosophy of the Internet has proved to be robust enough to stand the huge growth of the number of Internet hosts and applications during the last 30 years. Among the solutions that contributed to the success of the Internet, an important role is played by the congestion control algorithms that have protected the Internet from collapsing under the vastly increased load [1]. These principles have inspired the engineering design of TCP/IP protocols that is used in all current operating systems, as well as in the design of mobile and wireless technologies. Along this road, Transmission Control Protocol (TCP) and its numerous variants have become the norm for the provision of a reliable end-to-end data transmission. TCP carries more than 90% of today's Internet traffic and constitutes 80% of the total number of flows in the Internet [2]. Therefore, TCP is also widely used over wireless networks, as a significant majority of connections over such networks are using the Internet at some point in the end-to-end communication path. As the take-up of wireless technologies further proliferates, it is expected, for compatibility reasons, that TCP will be used much more often over wireless networks.

Using TCP—that was originally designed for wired networks—over wireless networks is especially challenging. The characteristics of wireless networks such as random and burst packet loss, intermittent connectivity, and highly variable shared media can mislead the TCP congestion control algorithm and degrade its performance dramatically. These challenging issues have inspired a large body of research over the last few years, of which an overview is given in the thesis.

1.1 Scope of the Work

TCP has been designed for wireline networks and its shortcomings over wireless networks, such as throughput degradation due to random losses and intermittent connectivity, have been the subject of a large volume of research investigations over the last few years. Various approaches have been taken to overcome these shortcomings. For example, several research efforts have proposed modifications to the transport layer, which involve a non-trivial and time-consuming standardisation problem. Another school of thought has emerged exploring the potential to enhance the algorithms of lower layers to improve the end-to-end performance of TCP. Moreover, recent measurements of Internet traffic show a wide diversity among the characteristics of the end-to-end TCP connections that can be used for the enhancement of the lower layer algorithms.

In the thesis, a set of techniques are proposed to enhance the performance of the end-to-end wireless communications using TCP for the transport layer. These techniques use information of the TCP connections such as Round Trip Time (RTT), congestion window (cwnd), and the flavour of TCP at the end host, thus making the wireless link-layer algorithms highly adaptive to these parameters. Novel algorithms are proposed at the link-layer to dynamically assign the parameters of Forward Error Correction (FEC), Automatic Repeat reQuest (ARQ), and wireless resource allocations. Therefore, using the end-to-end RTT as well as knowledge of the various flavours of TCP flows that are competing over the same wireless link, the persistency of the ARQ algorithm and the code rate of the FEC are set dynamically during the lifetime of each connection. The power and subcarrier allocation is also studied at the medium access control protocol that uses Orthogonal Frequency Division Multiple Access (OFDMA). In that respect, a novel joint downlink-uplink resource allocation algorithm is proposed, using the steady state throughput of each TCP flow.

The performance of the proposed schemes is investigated thoroughly. Various figures of merit such as end-to-end throughput, fairness among TCP flows, and end-to-end delay are used to present the efficiency of the proposed algorithms in wireless networks. In addition, network level simulation scenarios are performed in OPNET modeler to study the effect of the proposed algorithms on the real TCP implementations. The results show significant overall improvement in the performance of TCP over wireless.

1.2 Contributions of the Thesis

The contributions of this thesis lead to the design of a smart wireless link-layer that is TCP-aware. The design issues of this link-layer are discussed in three main chapters that are organised in a top-down fashion of the packet transmission process.

The first is to assign the coding scheme, second is to adapt the local retransmission policies, and the last is the allocation of actual wireless resources such as power and subchannel prior to data transmission. These proposals that are elaborated in chapters 3-5 of the thesis are as follows:

- The first proposal discussed in Chapter 3 is an adaptive FEC rate allocation at the link-layer that attempts to provide fairness among end-to-end competing TCP flows. The issue of fairness among TCP flows becomes critical when these flows are based on various TCP flavours that react differently to wireless and random losses. Therefore, an optimisation framework is presented that aims to maximise fairness among these heterogeneous TCP flows.

- A TCP-aware dynamic ARQ mechanism is presented in Chapter 4 that aims to prioritise packet (re)transmissions to avoid timer expiry at the transport layer, thus improves the performance of TCP. Moreover, this scheme avoids extra retransmissions of the already expired packets by TCP that results in more efficient channel utilisation. The delay analysis of the priority queuing policy used to design the dynamic ARQ mechanism shows that the queuing delay remains unchanged.

- Resource allocation scheme in OFDMA-based wireless is proposed such that, subcarrier/power allocation is performed in a TCP-aware fashion. The resource allocation problem is detailed over downlink aiming to provide more balanced throughput towards TCP, thus the achieved fairness among TCP flows is enhanced. Moreover, the joint uplink-downlink resource allocation scheme is proposed to address the problem of scarce availability of wireless resource over uplink. Considering that limited availability of bandwidth in the uplink, can degrade the performance of TCP over downlink, this problem is of special importance.

- Simulation investigations are performed within the OPNET modeler, which provides an opportunity to extensively study the effects of the proposed link-level algorithms on the performance of real TCP implementations.

- Finally, along with the simulation investigations, I contribute some models to the OPNET standard library. These contributions, which are also elaborated in Appendix B, are the adaptive wireless pipline stages that make the design of the smart link-layer possible, and the implementation of TCP Westwood that was utilised in the investigations of Chapter 3.

1.3 Structure of the Thesis

This thesis is structured as follows. After reviewing the related literature in Chapter 2, the main contributions of the thesis are discussed in chapters 3, 4 and 5. The concluding remarks together with some avenues of future research are detailed in Chapter 6. Finally, two appendices accomplish the technical contributions, in which the analytical and simulation models are illustrated. Appendix A elaborates the concepts of the mathematical tools that are utilised for the investigations. In addition, Appendix B depicts a clearer picture of the simulation modelling, and the philosophy and reasoning behind selecting the simulation platform.

The publications that are related to the contributions of the thesis are as follows.

1. Toktam Mahmoodi, Vasilis Friderikos, Oliver Holland, A. Hamid Aghvami, “Improving the Performance of TCP over Wireless Asymmetric Links,” submitted to IEEE Trans. VT, August 2009.

2. Toktam Mahmoodi, Vasilis Friderikos, Oliver Holland, A. Hamid Aghvami, “Optimal Design of FEC for Fairness Maximization among TCP Flavors over Wireless Networks,” IET Communications, submitted January 2009, conditionally accepted August 2009, final acceptance November 2009.

3. Toktam Mahmoodi, Vasilis Friderikos, A. Hamid Aghvami, “Allowing Short-Lived TCP Sessions to Ramp-Up in Broadband Wireless Networks,” Broadband Wireless Access Workshop, GLOBECOM’09, Honolulu, Hawaii, USA.

4. Toktam Mahmoodi, Vasilis Friderikos, Oliver Holland, A. Hamid Aghvami, “Balancing sum rate and TCP Throughput in OFDMA-Based Wireless Networks,” submitted to ICC’10, under review.

5. Toktam Mahmoodi, Vasilis Friderikos, Oliver Holland, A. Hamid Aghvami, "TCP-aware Resource Allocation in OFDMA Based Wireless Networks," International Workshop on Cross-Layer Design (IWCLD'09), Palma de Mallorca, Spain, June 2009.

6. Toktam Mahmoodi, Vasilis Friderikos, A. Hamid Aghvami, "TCP-aware Resource Allocation in Wireless Networks via Utility Maximisation," WWRF 22nd, Paris, France, May 2009.

7. Toktam Mahmoodi, Vasilis Friderikos, Oliver Holland, A. Hamid Aghvami, "Cross Layer Optimization to maximize Fairness among TCP flows of different TCP flavors," GLOBECOM'08, New Orleans, LA, USA, November 2008.

8. Toktam Mahmoodi, Oliver Holland, Vasilis Friderikos, A. Hamid Aghvami, "Cross-Layer Optimization of the Link-Layer based on the Detected TCP Flavor," PIMRC'08, Cannes, France, September 2008.

9. Toktam Mahmoodi, Oliver Holland, Vasilis Friderikos, A.Hamid Aghvami, "Effect of heterogeneous TCP versions on HTTP performance in Wireless Networks," LTE-Advanced International Research Workshop (Poster presentation), Vodafone, Newbury, UK, July 2008.

10. Toktam Mahmoodi, Oliver Holland, Vasilis Friderikos, A. Hamid Aghvami, "Cross-Layer Optimization of the Link-Layer based on the Detected TCP Flavor," WWRF 20th, Ottawa, Canada, April 2008.

11. Oliver Holland, Toktam Mahmoodi, Vasilis Friderikos, A. Hamid Aghvami, "Cross-Layer Optimization: Network Layer Involvement," WWRF 20th, Ottawa, Canada, April 2008.

12. Toktam Mahmoodi, Vasilis Friderikos, Oliver Holland, A. Hamid Aghvami, "Cross-Layer Design to Improve Wireless TCP Performance with Link-Layer Adaptation," IEEE VTC 2007 Fall, Baltimore, MD, USA, October 2007.

Chapter 2

Background Study

In this chapter, a full overview of TCP and its evolution over heterogeneous networks is given. The proliferation of wireless networks has significantly increased the challenges of using TCP over such networks. The organisation of this chapter, similar to the structure of the thesis, is based on the top-down approach. The structure of this chapter and where the actual contribution of the thesis lies, is depicted in Figure 2.1.

Section 2.1 gives an overview of the transport layer and its functionality. TCP as the most commonly-used transport protocol over the Internet, UDP, TFRC and DCCP as the other variants of transport layer designed for real-time applications are introduced. As the focus of this thesis is TCP, some variants of TCP that are proposed to enhance its performance over the heterogeneous networks are detailed. In Section 2.2, the trend of wireless technologies is elaborated, thus emergence of new generations of wireless networks and the arising challenges are discussed. After explaining the problematic behaviour of TCP over wireless networks in Section 2.3, the state of the art solutions are detailed. Section 2.4 shed the light on the main topic of the thesis and explain the cross-layer design proposals aiming to enhance the performance of TCP over wireless networks. Finally the methods to transfer the cross-layer information are also discussed in this section.

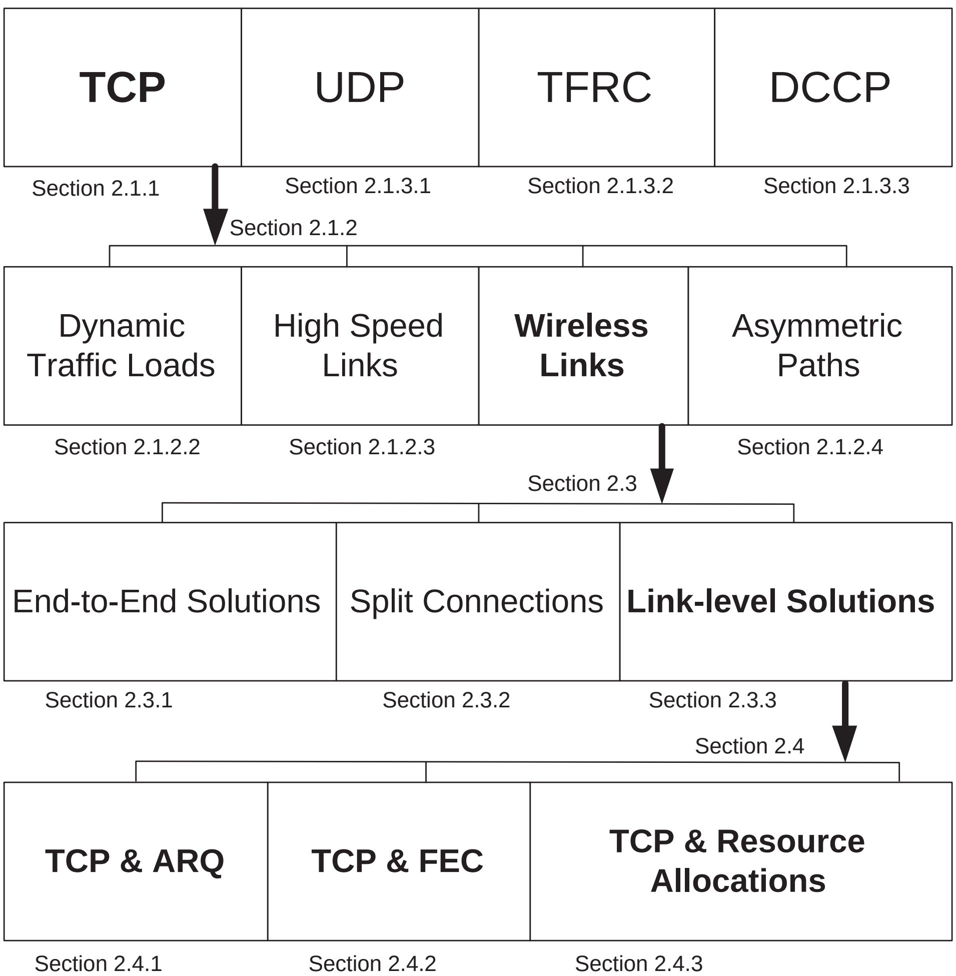


Figure 2.1: The structure of Chapter 2.

2.1 Transport Layer

Transport layer is a group of methods and protocols within a layered architecture of network components, which is responsible for encapsulating application data blocks into data units—datagrams and segments—suitable for transfer to the network infrastructure for transmission to the destination host, or managing the reverse transaction by abstracting network datagrams and delivering their payload to an application. Thus the protocols of the transport layer establish a direct, virtual host-to-host or end-to-end communications transport medium for applications and therefore also referred to as transport protocols [3].

2.1.1 Transmission Control Protocol (TCP)

TCP is designed to provide reliable end-to-end connection-oriented transport protocol over an unreliable internetwork—an internetwork differs from a single network mainly because different parts may have various topologies, bandwidths, delays, packet sizes and many other parameters. TCP is designed to dynamically adapt to the above mentioned properties of the internetwork [3]. The original specification for TCP is RFC 793 [4], although some errors have been corrected in the later RFCs. Currently, TCP is the default transport layer used in the Internet to provide reliable end-to-end communications and carries more than 90% of today's Internet traffic and 80% of the total number of the flows in the Internet [2; 5].

2.1.1.1 TCP Connection Establishment and Termination

As mentioned above, TCP is a connection-oriented protocol, thus prior to data transmission between two hosts, a connection needs to be established. In order to establish a new connection, the requesting end sends a segment calls SYN. The receiver TCP acknowledges the received SYN while responding with its own SYN segment. Upon reception of the ACKnowledge (ACK) and the SYN segments, sender

TCP establishes the connection, and also responds to the receiver TCP by sending the corresponding ACK. Receiver TCP also establishes the connection by receiving this ACK. These three segments complete the connection establishment that is often called three-way handshake.

Termination of the TCP connection can be initialised from each host, by sending a FIN segment. The other end ACKs the receipt of FIN segment, and sends its FIN segment to terminate the connection also from its own side. This FIN segment will be ACKed from the host that initiates the termination. Since TCP connection has a flow of data in both directions—it is bi-directional—, each direction must be terminated independently. Clearly, other events such as time out can also cause the connection to be shut down.

TCP connections, as bi-directional connections, require ACK from the receiver for the transmitted data to achieve the reliability. Therefore, timers are required in TCP to clock the transmitting data and to keep track of the received ACK for each sent packet. These timers are set based on the actual time each packet requires to travel over its corresponding end-to-end path.

2.1.1.2 TCP Timer Management

TCP continually measures the time an ACK packet takes to return, in order to distinguish the packets that have not reached the destination, thus retransmitting those packets. TCP maintains an averaging window over this delay, namely RTT, and also estimates the deviation of RTT from this average, D_{RTT}. The RTT timing is used to calculate the required waiting time for an ACK packet to return.

Thus, another timer is set to be used for the purpose of timer expiry of the transmitted packet that is called Retransmission Time Out (RTO). This timer restarts by the packet transmission, and if the corresponding ACK does not return before its expiration, TCP assumes that the packet is lost and retransmits it. The following equation calculates RTO

$$RTO = RTT + 4 \cdot D_{RTT}. \tag{2.1}$$

Although Equation (2.1) is used to calculate the expiration timer—RTO—, investigations have shown that small RTO can cause spurious retransmissions. Thus, TCP sets the RTO in a conservative manner such that if its calculated value is smaller than one second, RTO is set to one second. The specification that addresses how to compute the RTO is detailed in RFC 2988 [6].

2.1.1.3 TCP Congestion Control Algorithm

In October 1986, the first series of Internet "congestion-collapse" happened, which affect data throughput dramatically. For example, it has been reported that the throughput between two sites of the university of Berkeley was dropped from 32 kbps to 40 bps [1]. Jacobson was the first to study the impact of retransmissions on the throughput, based on experiments with the congested wired networks that lead to the base-line of the TCP congestion control algorithm [1].

Although the network layer also attempts to manage congestion, TCP has the major role in the congestion control, mainly because the clear solution to congestion is to slow down the data rate. The congestion control and avoidance algorithms of TCP are based on the notion that the network is a black-box, thus they operate by gradually increasing the load on the network until the network becomes congested and a packet is lost. To provide such a functionality, TCP congestion control algorithm introduces two variables of slow start threshold (ssthresh) and cwnd.

The cwnd represents number of packets that are allowed to be transmitted without getting acknowledged. The newly established TCP connection, initialises the cwnd with one segment equal to the Maximum Segment Size (MSS) and increases this value each RTT, which is the clocking of received ACK packets. When cwnd is equal to ssthresh, the increase rate in cwnd slows down. The size of cwnd can also be bounded by the maximum window size requested from the receiver TCP, which is called receiver's advertised window. The evolution of cwnd variable for different variants of TCP is further discussed in the next section.

As mentioned above, TCP gradually increases its transmission rate until a packet is lost, which is interpreted as congestion. On the other hand, treating loss as an indication of congestion in the network is appropriate for pure best-effort data traffic, with little or no sensitivity to delay. Although TCP congestion control mechanism added some techniques—such as Fast Retransmit and Fast Recovery—to minimise the effect of losses on the TCP throughput, these mechanisms do not consider delay requirement of non-elastic application. To this end, in conjunction to IETF efforts, significant amount of research has been carried out to design various flavours of TCP that perform well under the constraints and requirements of the new systems and applications.

2.1.1.4 TCP Enhanced Congestion Control

As the Internet grows both in terms of number of users and the amount of data traffic, many modifications are introduced to TCP. Therefore, the goal in design of the various versions of TCP is to provide more flexibility and scalability. In other words, the enhanced versions of TCP can provide better performance in the occurrence of random losses, in the dynamic traffic loads, and over the large bandwidth-delay product paths. However, a new variant of TCP, not only should enhance its performance under the new constraints, but also should be fair and friendly towards other TCP flows. Firstly, RTT fairness should be satisfied, so as to provide fairness among TCP flows when the competing flows have different RTTs. Secondly, TCP fairness should be considered, such that the new protocol must not reduce the performance of other regular TCP flows competing on the same path. After discussing the baseline design of TCP flavours in the next section, few more recent and well-used TCP versions are also introduced.

2.1.2 TCP Variations and Flavours

TCP must cope with the different transmission media crossed by the Internet traffic. By the increased heterogeneity of the Internet, this task is becoming more and more difficult. Issues such as, high speed links, long delay paths, shared and dynamic medium, lossy links, and asymmetric paths, are among the challenges. In this respect an overview of the performance of TCP over heterogeneous networks is given in [7], where some challenges and solutions are discussed. In this section, starting from the base line algorithms of the TCP congestion control, some state of the art solutions are detailed.

2.1.2.1 Baseline TCP's Congestion Control

TCP Tahoe refers to the congestion control algorithm that includes **slow start** and **congestion avoidance** phases [1]. TCP Tahoe suggests that whenever a TCP connection starts or restarts due to a packet loss, slow start procedure should be called. Slow start initiates the cwnd at the sender side to one, and after each received ACK, this window is doubled. Therefore, the cwnd is increased exponentially until a packet is lost or the cwnd reaches the ssthresh. In the first event the slow start is restarted and in the second event, TCP enters congestion avoidance phase.

In congestion avoidance phase, TCP Tahoe uses Additive Increase, Multiplicative Decrease (AIMD). In the occurrence of a packet loss, the ssthresh is set to the half of the current cwnd, and afterwards cwnd is set to one and slow start phase is restarted. TCP Tahoe detects a lost packet upon the expiration of the RTO. Therefore, a complete time out interval is required to detect a packet loss, which is the main drawback of TCP Tahoe. This issue has been addressed later in TCP Reno as the **fast retransmit** procedure.

TCP Reno, which is known as the base line of the modern TCPs, incorporate another notion of packet loss—congestion—into the Tahoe congestion control algorithm, which is Duplicate ACK (DupACK). TCP Reno generates an immediate ACK—

a DupACK—when an out of order packet is received. This DupACK, notifies that the packet in sequence is either delayed or lost in the network. Upon reception of the three DupACKs, TCP Reno suggests that there is a high probability of a packet loss, thus it enters the fast retransmit phase and retransmit the lost packet—without waiting for the time out. Afterwards, **fast recovery** procedure is performed, in which the ssthresh is set to the half of the current cwnd value, but not less than two, also cwnd is set to the ssthresh plus three, which inflates the cwnd by the number of packets received by the other host after the lost packet. In this phase, reception of each DupACK increases the cwnd by one until a fresh ACK is received. The time that takes for the new ACK to return, is at least one RTT. This received ACK not only acknowledges the retransmitted data, but also all the segments transmitted afterwards. Then, TCP reduces its cwnd to ssthresh, exits the fast recovery, and continues in congestion avoidance phase instead of restarting in the slow start [8].

The rational behind the implementation of fast retransmit and fast recovery algorithms is to allow higher throughput in the event of moderate congestions [9]. Although TCP Reno performs well in the presence of small number of packet loss, it can not be easily recoved from losing more than one packet in a window. Hence, TCP NewReno introduces some modifications to the fast recovery of Reno, in order to detect multiple packet losses, which is called partial ACK [10].

In the case of a single drop, the received ACK for the lost packet will acknowledge all the packets transmitted before fast retransmit was entered. However, when there are multiple packet losses, the ACK for the retransmitted packet will acknowledge some but not all of the packets transmitted before the fast retransmit. This ACK is called partial ACK. In this case, TCP NewReno retransmits the first unacknowledged packet, and deflates the cwnd by the amount of acknowledged data. Thereby, the performance of TCP over lossy links can be significantly improved.

Including the Selective Acknowledgement (SACK) option in TCP header allows the receiver to report a non-sequential received data. The SACK option can significantly improve the performance if there is a large number of packet losses per transmission window [11]—e.g., if there are burst-losses. TCP SACK option was formally expressed in RFC 2018 [12], and later, its extensions are discussed in RFC 2883 [13]. The SACK option allows a receiver to specify, in acknowledgements, the whole blocks of packets which have been received successfully. However, the options part of a TCP header is only large enough to allow for a maximum of four SACK blocks. Generally, not more than three SACK blocks are used in the TCP header, as SACK is often used in conjunction with the Timestamp option in the TCP header.

2.1.2.2 TCP Over Dynamic Traffic Loads

In the above discussed TCP versions, the cwnd is blindly halved in the event of packet loss. This can constrain the higher throughputs to be achieved on the end-to-end path. On the other hand, in shared medium access the available bandwidth for a TCP flow is highly variable depending on the channel utilisation and medium access protocol dynamics. If a sudden change in available bandwidth occurs, TCP may be too slow to converge to this bandwidth. Therefore, another category of TCP congestion control algorithms are introduced that attempts to alter the cwnd depending on the available bandwidth, which can be estimated e.g. from the rate of ACK packets. Among these proposals [14], TCP Vegas and TCP Westwood, and their congestion control algorithms are detailed here.

In 1994, Brakmo, O'Malley and Peterson proposed a new TCP called Vegas, and they explained that Vegas can achieve between 40% to 70% better throughput comparing with Reno [15]. The principle of TCP Vegas is based on the assumption that in a non-congested network, the actual flow rate should be close to the expected rate. Therefore, the difference between the expected and the actual flow rate, D, is used to estimate the available bandwidth, and the cwnd is updated accordingly. Given two thresholds, e.g., α and β corresponding to too little and too much data in the network, TCP Vegas increases or decreases the cwnd linearly at the next RTT accordingly.

$$cwnd = \begin{cases} cwnd + 1 & D < \alpha, \\ cwnd + 1 & D > \beta, \\ cwnd & otherwise. \end{cases} \tag{2.2}$$

Using this approach TCP Vegas detects and utilises the extra bandwidth whenever it becomes available without congesting the network. Thereby, Vegas avoid using loss as the notion of congestion, instead uses its observation from the utilised bandwidth to recognise congestion.

TCP Westwood is a sender side only modification to TCP Reno—or NewReno—that can provide performance improvement over the paths with large bandwidth-delay product, transmission errors, and dynamic loads [16]. The key idea in TCP Westwood is to estimate, at the TCP sender, the efficient packet rate of the connection by monitoring the ACK reception rate, i.e. if ACKs are being returned at a certain rate, then packets are getting to the receiver at that same rate hence the network can support that rate. The Eligible Rate Estimation (ERE), which is calculated by low-pass filtering the rate of incoming ACKs, allows the sender TCP to intelligently set the cwnd and ssthresh at the event of loss. The ERE is computed over the time period T_k that depends on the congestion level, and is calculated as follows.

$$T_k = RTT \cdot \frac{\frac{cwnd}{RTT_{min}} - RE}{\frac{cwnd}{RTT_{min}}} \tag{2.3}$$

where RTT_{min} is the minimum RTT value of all acknowledged packets in a connection. In this equation, the expected data rate when there is no congestion is given by $\frac{cwnd}{RTT}$, while RE is the achieved rate computed based on the amount of acknowledged data during the latest RTT. In response to a packet loss as detected by DupACKs, TCP Westwood sets the cwnd and ssthresh to the ERE. Moreover, TCP Westwood shows that it performs friendly with the other versions of TCP—thus TCP fairness is satisfied. Although Westwood uses the dynamicity of bandwidth to update the cwnd, the notion of congestion for this TCP flavour is still based on packet loss.

2.1.2.3 TCP Over High Speed Links

Another issue arises in high speed and long distance networks, where the TCP cwnd needs to be large enough to fully utilise the available resources. However, the conservative approach of TCP, e.g., TCP Reno, in increasing the cwnd can not satisfy this requirement. An example that illustrates a more clear picture is detailed in [17], where it is pointed out that over a 10 Gbps link with 100 ms delay, it will roughly take one hour for a standard TCP flow to fully utilise the link capacity, if no packet is lost or corrupted. The proposals for the high speed TCP can be classified into two categories.

The first class of solutions, applies changes to the increase/decrease rate of the cwnd in congestion avoidance phase—it has been shown that the slow start phase is enough aggressive in updating the cwnd. Among the proposals in this area, High Speed TCP (HSTCP) [17] and Scalable TCP (STCP) [18] are typical examples. STCP alters the AIMD with the Multiplicative Increase, Multiplicative Decrease (MIMD), thus increases the cwnd by 0.01·MSS on every received ACK and reduces cwnd to its 0.875 times in the event of loss. On the other hand, HSTCP maintains the AIMD but some changes are applied such that, as cwnd increases, the decrease parameter reduces, while the increase parameter grows accordingly. These proposals are far more aggressive than for example TCP Reno, thus they suffer from the RTT unfairness as well as the TCP unfairness.

Another class of high speed protocols, like FAST TCP [19], introduces a new design of the congestion control algorithm that takes the variants of RTT as the indicator of congestion, namely delay-based congestion control. The key idea of the delay-based congestion control algorithms is that the increase of RTT is considered as an early congestion notification, thus sending rate is reduced to avoid buffer overflow. Moreover, FAST TCP incorporates multiplicative increase if the size of buffer occupied by the connection at the bottleneck is far less than some pre-defined threshold, and switches to linear increase if it is close to that threshold. Although delay-based protocols show that they can achieve better properties such as higher utilisation of the available bandwidth, and better RTT fairness, they perform poorly in competing with the loss-based protocols such as Reno. Since the delay-based flows attempt to maintain a small number of packets in the bottleneck queue, they will stop increasing their sending rate when the delay reaches some value. However, the loss-based flows will not react to the increase of delay, and continue to increase their sending rate, which cause the delay-based flows to further reduce their cwnd.

Compound TCP (CTCP) is a proposed algorithm by Microsoft that combines the delay-based and loss-based congestion control algorithms. Considering that CTCP is implemented as part of the Windows Vista and Window Server 2008 TCP stack, it is worthwhile to be discussed here. In CTCP, a scalable delay-based component is added into the TCP Reno loss-based congestion control algorithm. This delay-based component can rapidly increase the transmission rate when network is under utilised, but gracefully retreat in a congested network [20].

2.1.2.4 TCP over Asymmetric Paths

The effect of link asymmetry on the performance of TCP is widely studied in the literature. Limited available bandwidth and congestion on the reverse path break down the principle of ACK clocking, and may cause an increase in the RTT [21]. Thus, TCP throughput on the forward path is degraded. Among several research works that explore this issue, some require explicit support from routers or middle nodes, whereas others are end-to-end schemes. For example, ACK congestion control [22] is one of these solutions that attempts to reduce the sending rate for ACK traffic, with the assumption that the reduction in ACK rate may help to cut the congestion itself.

2.1.3 Other Transport Protocols

2.1.3.1 User Datagram Protocol (UDP)

User Datagram Protocol (UDP), which is formally defined in RFC 768 [23], is the well-used transport protocol over the Internet for delay sensitive applications that is a much simpler protocol comparing with TCP. UDP provides a connectionless protocol, thus does not set up a dedicated end-to-end connection. In addition, communication is achieved by transmitting information in one direction from source to the destination, hence the reliability and keeping track of the packets order can not be achieved. In spite of all the disadvantages, UDP has the benefit of being simple, hence it has been so far one of most commonly-used transport protocol for real-time applications. For example in VoIP, loosing a packet is only a slight degradation in quality, although using a reliable transport layer may cause large delays if lost packets are retransmitted. Applications that require congestion control and utilise UDP, should implement the congestion control algorithm within their applications.

2.1.3.2 TCP-Friendly Rate Control (TFRC)

TCP-Friendly Rate Control (TFRC) is a congestion control mechanism that is reasonably fair when competing for bandwidth with TCP flows, but has a much lower variation of throughput over time, comparing with TCP. Therefore, it is more suitable for applications such as telephony or streaming media where a relatively smooth sending rate is of importance. TFRC that is formally defined in RFC 3448 [24], utilises the throughput equation of TCP for its congestion control mechanism. By using TCP throughput to update the cwnd, fairness in competition with TCP flows is provided. The throughput of TCP can be expressed as a function of loss rate and RTT—more details on these equations are given in Section 2.1.4. Despite smoother throughput, TFRC responds slower than TCP to the changes in the available bandwidth. Therefore, for applications that simply need to transfer as much data as possible TFRC is not recommended.

2.1.3.3 Datagram Congestion Control Protocol (DCCP)

Datagram Congestion Control Protocol (DCCP), which is a more recent proposal, can be used as an adaptive transport protocol. It is a transport protocol that provides bidirectional connections of congestion-controlled unreliable datagrams. DCCP is formally defined in RFC 4340 [25], and is suitable for applications such as streaming. Mainly because it can benefit from control over the tradeoffs between delay and reliable in-order delivery. Thus, DCCP allows applications to choose from a set of congestion control mechanisms. One alternative, TCP-like Congestion Control, halves the cwnd in response to a packet loss, similar to TCP. Applications using this congestion control mechanism will respond quickly to the variations in the available bandwidth, but they must tolerate the abrupt changes in the TCP-like cwnd. The second alternative, TFRC, minimises abrupt changes in the sending rate while maintaining longer-term fairness with TCP. The standard of DCCP is also open to the other alternatives of the congestion control mechanisms [25]. Moreover, Linux has an implementation of the first release of DCCP in kernel version 2.6.14—released October 28, 2005.

2.1.4 TCP Throughput Modelling

As mentioned earlier, TCP steady state throughput can be expressed as a function of the Packet Error Rate (PER) and the RTT. This function can be different for various TCP flavours, as the reaction of each of these flavours to packet loss is different. In this section, the throughput expressions of TCP Reno, TCP NewReno, TCP Reno with the SACK option, and TCP Westwood are detailed. These variants of TCP are used in the investigations of the thesis. The expressions of TCP throughput is based on the steady-state throughput that a TCP flow can achieve, thus TCP connections are assumed to be sufficiently long-lived.

2.1.4.1 Modelling TCP Reno

Under the assumption of independency of packet losses between rounds, and that the sender's rate is not limited by the receiver's advertised window, a closed form for TCP Reno throughput in the steady state is proposed in [26] and revised in [27]. This gives the TCP Reno throughput, $B_R\left(p\right)$, as follows,

$$B_R\left(e\right) = MSS \cdot \frac{\frac{1-e}{e} + E\left[W\right]}{\overline{RTT}\left(\frac{b}{2} \cdot E\left[W\right] + b + 1\right)}. \;\; bytes/s \tag{2.4}$$

In the above expression, b represents the number of download packets that each acknowledgement applies to—in the most commonly used implementations of TCP, the default value of this is one—, e is the packet loss probability, and $\overline{RTT}$ is the average value of the RTT. Finally, $E[W]$, the expectation of the cwnd, is defined by,

$$E\left[W\right] = -\frac{3b-2}{3b} + \sqrt{\frac{8\left(1-e\right)}{3be} + \left(\frac{3b-2}{3b}\right)^2}. \tag{2.5}$$

2.1.4.2 Modelling Enabled SACK option

In the presence of the SACK option in TCP Reno, the throughput model is presented and validated in [28], where its Reno model presents similar results to [26]. Given the assumption that a loss happened in the jth packet of the round (one RTT) when the flow has a cwnd of h, the throughput expression is as follows,

$$B_{SACK}\left(e\right) = \frac{MSS}{e} \cdot \left(\frac{2}{c_{wm}(c_{wm}+1)} \sum_{h=1}^{c_{wm}} \sum_{j=1}^{h} \left(2 + r(n) + r(a,n)\right) \cdot \overline{RTT}\right)^{-1}. \;\; bytes/s \tag{2.6}$$

In Equation 2.6, it is assumed that the cwnd varies uniformly between 1 and c_{wm}, maximum cwnd. The number of rounds that is required to transmit n packets is denoted by $r(n)$. Finally, with the initial cwnd value of n, it takes $r(a,n)$ rounds to transmit a packets in the congestion avoidance phase.

2.1.4.3 Modelling TCP NewReno

An analytical model for the TCP NewReno throughput is proposed in [29], where the extra assumption of correlated losses to the above applies. Given $\bar{\delta}$ the average number of segments lost per loss event, the throughput model details as follows,

$$B_{NR}(e) = MSS \cdot \frac{1 - e + E[W]}{\overline{RTT} \cdot (1/2E[W] + b(1/2E[W] + 1) + 1)}, \quad bytes/s \tag{2.7}$$

where,

$$E[W] = \begin{cases} -\frac{3b+2\bar{\delta}}{3b} + \sqrt{\frac{8(1-e)}{3eb)} + \frac{4\left(\bar{\delta}^2+\bar{\delta}-2\right)}{3b} + \left(\frac{3b+2\bar{\delta}}{3b}\right)^2}, & \bar{\delta} < E[W], \\ -\frac{3b-1}{3b+1} + \sqrt{\frac{8(1-e)}{e(3b+1)} + \left(\frac{3b-1}{3b+1}\right)^2}, & \text{otherwise.} \end{cases} \tag{2.8}$$

2.1.4.4 Modelling TCP Westwood

The analytical throughput model for TCP Westwood, is proposed in [30]. In this model, it is assumed that the system is always in the congestion avoidance phase, and that only a single packet loss occurs in each round. The packet transmission rate, given by μ, is assumed to be constant. If the burst at which the pipe capacity is reached is denoted by b_{k^*}, the value of k^* is determined from $k^* = C - W_0 + 1$, where C is the pipe capacity and W_0 is the size of the initial congestion window. Assuming that the buffer can hold up to B packets, the packet number dropped due to buffer overflow, n_{of}, can be formulated as $n_{of} = s_{k^*} + 2(C + B)$. The determination of s_k that is the first packet of burst b_k is given by,

$$s_k = 1 + (W_0 - 1)(k - 1) + \frac{k(k-1)}{2}. \tag{2.9}$$

The number of the bursts that contains packet number n is denoted by k_n, and the offset of the packet in burst b_{k_n} by r_n. Thus, k_n and r_n can be expressed as follows,

$$\begin{aligned} k_n &= \left\lfloor -W_0 + \tfrac{3}{2} + \sqrt{W_0^2 - W_0 - \tfrac{7}{4} + 2 \cdot n} \right\rfloor, \\ r_{k_n} &= n - s_{k_n}. \end{aligned} \tag{2.10}$$

The instant in which the nth ACK is received is then given by

$$t\left(W_0, n\right) = \begin{cases} T_{k_n} + \frac{r_n}{\mu}, & n \leq s_{k^*}, \\ T_{k^*} + \frac{n - s_{k^*}}{\mu}, & n > s_{k^*}. \end{cases} \tag{2.11}$$

The probability $p_{W_0,n}$ that packet n is dropped is expressed as

$$p_{W_0,n} = \begin{cases} e\left(1-e\right)^{n-1}, & n < n_{of}\left(W_0\right), \\ \left(1-e\right)^{n-1}, & n = n_{of}\left(W_0\right). \end{cases} \tag{2.12}$$

Finally, given π_W the asymptotic probability of the initial congestion window size being W_0, the average throughput is expressed by,

$$B_W = MSS \cdot \sum_{W_0=2}^{C} \pi_W \sum_{n=1}^{n_{of}} \frac{n-1}{t(W_0, n)} p_{W_0,n}. \ bytes/s \tag{2.13}$$

2.2 Evolution of Wireless Networks

Despite the fact that traditional wireless and mobile networks were targeted at voice, wireless data traffic increased dramatically over the past few years. In other words, wireless communications have become pervasive. Moreover, the recent report from Cisco on Global "Mobile Data Traffic Forecast Update" predicts a drastic increase in the number of mobile Internet users and the amount of wireless data traffic [31]. New generations of mobile wireless networks aim not only to support high data rates, but also to provide some degrees of Quality of Service (QoS) for e.g. multimedia services.

The first generation of cellular systems became available in 1983. In Europe, Total Access Communications System (TACS) was introduced with 8 kbps data rate, and using of the Frequency Division Multiple Access (FDMA) as its multiplexing technique. This network was soon replaced by Global System for Mobile communications (GSM), as the 2nd Generation (2G) of mobile networks, which is based on Time Division Multiple Access (TDMA) and benefits from digital channels.

GSM network that aims to support voice and some limited data communications, was first launched in 1991 in Finland. Afterwards, the move into the 2.5G world began with General Packet Radio Service (GPRS) that adds the packet switching protocols to GSM.

To this end, 3rd Generation (3G) of mobile networks have emerged with few alternatives, aiming to provide data services. International Mobile Telecommunications (IMT)-2000, which is the global standard for 3G wireless communications, provides a framework for worldwide wireless access by linking the diverse systems of terrestrial and/or satellite based networks. It exploits the potential synergy between digital mobile telecommunications technologies and systems for fixed and mobile wireless access systems. Moreover, Universal Mobile Telecommunications System (UMTS) as an umbrella term for the 3G radio technologies, which is developed within 3GPP, and its evolved version High-Speed Downlink Packet Access (HSDPA), are the current 3G mobile cellular systems. These technologies could not accomplish the data rate and QoS requirements of the new applications.

While the efforts are continued by both the research and standardisation communities for the fourth generation of wireless networks, different alternatives such as, WiMAX, LTE, and LTE-Advanced have been pushed forward. In this section a brief introduction to these three wireless networks is given. In addition, WLAN which has enabled the local area networking, is also discussed here.

2.2.1 Wireless Local Area Network (WLAN)

Wireless Local Area Network (WLAN) that is based on the IEEE 802.11 standard [32], provides short range wireless networking using the unlicensed 2.4 or 5.3 GHz unlicensed radio band. In the past few years, some enhancements have been introduced to the WLAN specifications, that result in e.g. IEEE 802.11a that uses Orthogonal Frequency Division Multiplexing (OFDM) technique, IEEE 802.11n that supports Multiple Input, Multiple Output (MIMO), and IEEE 802.11s for mesh networking. Moreover, WLAN access the medium using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to minimise the collision.

2.2.2 Worldwide inter-operability for Microwave Access (WiMAX)

WiMAX provides point-to-multipoint wireless communications based on the IEEE 802.16 specifications [33]. It is described as “a standard-based technology enabling the delivery of last mile wireless broadband access” by its forum. The radio frequency band was initially defined in the range 10 to 60 GHz, and afterwards IEEE 802.16a [34] adds specifications for the 2 to 11 GHz. Further enhancements have been added to the standard in IEEE 802.16e [35], in which the use of OFDMA as the access method, and support of MIMO and mobility have been introduced. The IEEE 802.16e with these amendments is among the candidates of the fourth generation of mobile networks.

2.2.3 Long Term Evolution (LTE)

Long Term Evolution (LTE) that is the 3GPP specification for the fourth generation of mobile networks, also referred to as Evolved UMTS Terrestrial Radio Access (E-UTRA). LTE represents a step forward for the wireless communications, targeting order-of-magnitude increases in the bit rates with respect to its predecessors by means of wider bandwidths and improved spectral efficiency. Beyond the improvement in bit rates, LTE aims to provide a highly efficient, low-latency, packet-optimised radio access technology offering enhanced spectrum flexibility. The OFDM technique has been selected for the downlink and Single Carrier-FDMA (SC-FDMA) for the uplink. The downlink supports data modulation schemes QPSK, 16QAM, and 64QAM and the uplink supports BPSK, QPSK, 8PSK and 16QAM. At the network layer, a flatter architecture is being defined that represents the transition from the existing UTRA network, which combines circuit, and packet switching, to an all-IP system.

2.2.4 Long Term Evolution (LTE)-Advanced

In 2008, 3GPP held two workshops, where the "Requirements for Further Advancements for E-UTRA" were discussed. The resulting technical report that has been published in [36], addresses the mobile systems whose capabilities go beyond those of IMT-2000, and called IMT-advanced. Some of the main objectives have been identified as interworking with other radio access systems, and enhanced peak data rates to support advanced services and applications—100 Mbps for high and 1 Gbps for low mobility users.

2.3 TCP over Wireless and Mobile Networks

In theory, transport protocol i.e. TCP should be independent of the underlying medium. However, when the underlying medium detracts from the reliable, wired media that TCP was originally designed to serve, it exhibits a number of shortcomings. On the one hand, wireless networks are characterised by random and high probability of errors and also intermittent connectivity. On the other hand, congestion control algorithms interpret packet loss as the indication of congestion. As discussed in section 2.1.2, in response to the packet loss, most of the TCP variants—loss-based congestion control—decrease the cwnd and the transmission rate accordingly, thus a dramatic degradation in TCP throughput can occur. Mobile users explicitly, can significantly affect the TCP throughput due to mobility and handoff that may cause frequent disconnections. As the RTO of TCP is doubled after each expiration, reconnection after few disconnection results in a very long waiting time to recover the previous transmission rate—this event is called a serial time-out at the sender TCP.

A significant amount of research have evolved over the past few years, aimed at improving the performance of TCP over wireless and mobile networks [37; 38; 39; 40]. Three schools of thought have emerged: the first class proposes the end-to-end solutions where the sender TCP is aware of the wireless link and changes should be applied to the transport-layer protocol, while the second and the third explore the potential to solve the problem locally at the wireless link, thus hide the unreliability of the wireless from the TCP. In this respect, the second category suggests to break the end-to-end connection to wired and wireless connections, and the third class attempts to enhance the characteristics of the link-layer, e.g. to increase the reliability at the link-level. These three categories are discussed here inline with the proposed solutions in the literature. The focus of this thesis is on the third class of solutions, thus it is further discussed in Section 2.4.

2.3.1 End-to-End Solutions

As discussed earlier in section 2.1.2, various enhancements are proposed to the baseline of TCP congestion control. Some of these proposals aim to improve TCP performance over the unreliable medium such as wireless. For example, TCP Westwood shows a significant performance enhancements in the event of random losses comparing with TCP Reno [16].

Using another notification for congestion—instead of the packet loss—can be used to avoid the misinterpretation of random losses with congestion. Some of these proposals are discussed in section 2.1.2 as delay-based congestion control algorithms. Explicit Congestion Notification (ECN) is also a well-used example of this approach, in which routers report congestion to the sender TCP using IP header [41]. It is assumed that the active queue management is deployed at the middle routers that allows routers to detect congestion before the queue overflows and loss occurred.

A similar approach to ECN is used in [42], in which a biased queue management scheme is established at the routers to distinguish congestion losses from random losses. The key idea of this protocol, entitled TCP Casablanca, is to de-randomize congestion losses so that the distribution of congestive losses differs from that of random wireless losses. To achieve this, TCP sender marks some of the sent packets; in the event of congestion, routers drop those packets first. The simulation results presented by the authors in [42] show that, TCP Casablanca identifies congestive losses with more than 95% accuracy and wireless losses with more than 75% accuracy.

2.3.2 Split TCP Connection

The main idea behind the split TCP connection is to isolate wireless related issues from the TCP connection on the wired part of the network. Given a split point, the end-to-end TCP connection can be divided into two separate connections: one between the mobile user and the split point (e.g. wireless base station), the other between the split point and the end-host. Among the existing proposals the followings are discussed here to cover the variants of the solutions: I-TCP, M-TCP, and TCP proxy.

The Indirect-TCP (I-TCP) is one of the early protocols to use the split-connection approach. It splits the end-to-end link from the mobile user to the end-host at the wireline-wireless border, thus wireless base station plays the role of the split point [43]. Afterwards, the wireless base station establishes a TCP connection with the end-host on behalf of the mobile user. If a handoff occurs during the life-time of this TCP connection, the new wireless base station takes over from the previous one. The other TCP connection is established between the wireless user and its station. This connection can be either a regular TCP connection, or a less complex transport layer.

Mobile TCP (M-TCP) is proposed based on the similar concept with I-TCP, while it mainly focused on the mobility issues. Therefore, frequent disconnection and power scarcity at the mobile are assumed to be the main challenges [44]. This protocol is a split connection protocol where the split point is at the “supervisor” host that controls several wireless base stations. The M-TCP protocol, assigns a fixed amount of bandwidth to each mobile node and performs the local error recovery. The regular TCP connection is established between the "supervisor" host and the end-host at the wired side, but a special protocol setup a connection between the mobile host and the supervisor host. Despite the complex architecture used by M-TCP, its advantage over I-TCP is that less amount of data are required to be transferred in the event of handoff.

TCP proxy locates a proxy at the split point, such that, first receives all the data from the sender and only then, forwards data to the receiver. The proxy can perform data compression before transmitting data to improve the performance. Clearly, TCP proxy results in a relatively long delay before receiving any data packets, because it takes long time to finish uploading the file to the proxy via the wireless connection. Despite this fact, it has been shown that TCP Proxy with data compression can outperform the other split connection solutions such as I-TCP [45].

One of the main concerns regarding the split TCP connection is that the end-to-end semantic of TCP is violated, thus TCP connection my not terminate gracefully when it is aborted in the middle. This issue is examined in [45], where experiments show that termination of the connection between the end-host and the split point can be delayed up to 12 minutes.

2.3.3 Link-Level Solutions

Link-Level solutions attempt to provide reliability at the link-level such that the transport layer can be shielded from the wireless events. The well-known examples of such reliable algorithms are link-level ARQ and link-level FEC or the hybrid scheme of Hybrid ARQ (HARQ). By local retransmission of the lost packet, ARQ mechanism can provides reliability at the link-layer. FEC technique can also increase the probability of packet delivery by adding some redundancy to the transmitting packet. Clearly, the degree of reliability at the link-layer depends on the persistency of the ARQ mechanism, and the extra redundancy of the FEC scheme, which will be discussed with more detail in Section 2.4.

One of the very early link-level solutions, is the Snoop protocol that introduces a Snoop agent at the wireless base station [46]. The snoop agent monitors every packet of the corresponding TCP connection in both directions and buffer the packets that have not yet been acknowledged by the receiver. Moreover, Snoop agent keeps a timer for the transmitted packets, thus packet loss can be detected either by the arrival of the number of DupACKs or by the expiration of its timer. Thereby, the DupACKs are suppressed and the retransmission is performed locally. Despite the advantage of suppressing DupACK packets that can avoid unnecessary fast retransmit by TCP, Snoop protocol can suffer from not being able to shield TCP from wireless losses.

The main advantage of employing a link-layer protocol for loss recovery is that it fits naturally into the layered structure of network protocols, and operates independently of higher-layers. On the other hand one of the main concerns is the possibility of adverse effect of such protocols on TCP, e.g., ARQ mechanism can cause out-of-order packets, or may increase the delay and lead to timer expiration.

As the focus of the thesis is on the link-level solutions with the cross-layer approach, this issue is further discussed in the next section. Moreover, some methodologies to transfer the cross-layer information between TCP and link-layer are given.

2.4 Cross-Layer Approaches towards TCP

Among the various solutions that have been proposed to enhance the performance of TCP over wireless networks, the cross-layer approach aiming to design a wireless link-layer with TCP awareness has received significant attention [47]. This category of solutions benefits from the fact that they fits into the natural design of the protocol stack. In other words, slight modifications to the wireless link-layer or sometimes adapting the parameter of the existing algorithms can improve the performance of the end-to-end connection at the end-user—mobile user— dramatically.

This tapestry of solutions attempts to design a smart link-layer such that its procedures and algorithms can be controlled in a TCP-aware manner. The well-known examples of these procedures are the modulation and coding schemes, the actual resource allocation schemes such as power and channel allocations, and the mechanism that provide reliability at the link-layer. The most commonly-used mechanism to provide reliability at the wireless link layer are the ARQ and FEC schemes.

2.4.1 TCP and Automatic Repeat reQuest (ARQ)

Retransmissions of the lost packets at the link-layer is handled via ARQ mechanism that is mainly characterised by its level of persistency in retransmission. ARQ mechanism can use different approaches to detect the packet loss among which the Negative ACK (NACK) and timer expiry are the most commonly-used. Thereby, ARQ transmits the packets over the wireless link and keeps the copy of transmitted packet until the packet is acknowledged. If the packet is not acknowledged within its corresponding timer, then ARQ retransmits the packet that is a similar approach to the TCP retransmission due to the time-out. In addition, NACK is sent by some variants of ARQ, if an out-of-order packet is received, thus sender can be informed of the loss before timer expires.

The number of (re)transmission attempts by an ARQ mechanism is called its persistency level. The persistency level of any ARQ mechanism can be classified into three categories: Perfectly-Persistence, High-Persistence, and Low-Persistence [48]. Consequently, these three classes of persistency, results in a Fully-Reliable, Highly-Reliable, and Partially-Reliable link-layer. On the other hand, ARQ retransmissions increase delay in the packet delivery, thus augmenting the level of reliability can amplify this delay that is not desirable for many applications. Thereby, the tradeoff between reliability and delay exists here in the design of an ARQ mechanism. This tradeoff is carefully discussed in RFC 3366 [48], and recommendations for the link-layer designers are given accordingly. Most of the link-layer designs choose one of the last two classes of high/low-persistence ARQ, and find the maximum number of (re)transmission attempts based on the constraints imposed by either wireless link or higher-layers.

As ARQ performs a similar functionality to TCP, the cross-layer interaction between ARQ and TCP and the effect of ARQ mechanism on TCP performance is discussed in numerous research works. On one hand, retransmission of the lost packet at the link-layer can avoid the congestion control algorithm of TCP to react and reduce its cwnd due to random loss. On the other hand, the latency that is introduced to the packet transmission by ARQ increases the RTT of TCP, and slows down the increasing rate of its cwnd that results in reducing the TCP throughput. In addition, the independency of these two retransmissions, may cause extra and not-required retransmissions, which could results in the packet re-ordering in TCP.

The authors of [49] design a cross-layer algorithm that dynamically adapts the maximum number of retransmissions in the ARQ protocol according to the end-to-end loss rate experienced by TCP. The key idea of this proposal is based on the experimental results showing that, when the packet loss probability experienced by TCP in the wireless link is of the same order of magnitude as the one experienced in the wired part of the end-to-end path, the performance of TCP is not affected by the wireless losses. In other words, implementing a reliable wireless link while TCP experiences significant packet loss on the wired part of the path, may have a negative impact on the wireless link utilisation without improving the TCP performance. A Markov model is developed to reproduce the ARQ protocol; thereby, TCP performance over UMTS is investigated using this scheme. The results reveal that, there is no need to have unlimited number of retransmissions at the ARQ protocol, as the proposed adaptive ARQ with limited number of retransmissions can achieve similar TCP throughput to that. Moreover, it has been shown that although the end-to-end TCP throughput is improved by the adaptive ARQ but there is no enhancement in terms of wireless efficiency such as the loss probability on the wireless link.

Given the assumption of having MIMO channels, TCP performance over ARQ is studied in [50], thus the tradeoffs of spectral efficiency and retransmissions from a TCP cross-layer standpoint is investigated. The results of this study show that under poor channel condition—SNR below 12dB—using space-time coding results in better performance than using ARQ mechanism. On the other hand, when channel conditions improve, using ARQ mechanism is preferred. Moreover, investigations reveal that the more reliable the system and the more correlated the channel, the less improvement the ARQ provides. However, in some cases, increasing the persistency of ARQ has a negative impact on the RTT of TCP and results in the degradation of TCP throughput.

The proposed solution in [51], avoids transmitting TCP ACK packets over the wireless link, and generate the ACKs locally at the wireless base station. This scheme that is called Link-Layer ARQ Exploitation TCP (LLE-TCP), improves the performance mainly by reducing the traffic over the wireless link. Moreover, the RTT of TCP is decreased by the local generation of ACKs, which can results in improving the TCP throughput. The simulation studies are performed in WLAN scenarios, where it has been shown that throughput can be increased by 20-100% depending on the packet size that is used.

2.4.2 TCP and Forward Error Correction (FEC)

The FEC scheme at the link-layer can improve reliability by creating redundant data from the original data sequence prior to transmission. This redundant data allow the receiver to correct a proportion of errors caused by corruption from the wireless channel, hence retransmissions are avoided. Increasing the redundancy improves the reliability of FEC, but also increases the channel load and the processing/reception delay. Therefore, the tradeoff between the wireless channel utilisation and delay on one side and the reliability degree on the other side is widely studied in the literature.

In [52], Barakat and Altman examine the channel utilisation and reliability tradeoff with respect to the throughput of TCP. They show how TCP performance varies as a function of the FEC rate, hence the optimal rate of FEC is selected so that TCP throughput gains the most. This FEC rate is a rate that improves the PER experienced by TCP such that, TCP throughput will be equal to the data transmission over wireless link. In addition, performance investigations in the presence of correlated losses derive two other conclusions. Firstly, it has been shown that an increase in the burstiness of the channel, even if the average error rate remains the same, reduces the efficiency of FEC, thus degrades the throughput of TCP. Secondly, using a large packet size is recommended, as the large packets show more resilient to the burst errors without adding to the FEC rate.

The research work presented in [53], brings the FEC scheme, ARQ mechanism and power allocations—as the three parameters that can control the transmission quality–into the unique framework, and defines a utility function to find the optimal balance among these three parameters. Through simulation results, it has been shown that increasing power, redundancy bits of FEC and retransmission attempts of ARQ, always improve the TCP performance by reducing packet loss, but these improvements are associated with extra cost. Therefore, a joint optimisation problem over cost and TCP throughput is defined to tune power, FEC and ARQ, and improve TCP performance.

Another approach is discussed in [54], where the main attempt is to find the optimal redundancy rate of FEC that enhances the latency of TCP packet transmission. The presented results in this paper reveal that, adding to the FEC rate firstly improve the latency but after certain threshold, it deteriorates TCP delay. Among various TCP flows, those with the smaller RTT, are more sensitive to this extra delay. In fact, this increase in the end-to-end delay is relatively less important when the link delay is large.

A cross-layer interaction between TCP and link-layer is proposed in [55], to protect TCP packets against losses on the wireless channel when the TCP congestion window size is below the bandwidth-delay product of the network. In this respect, the link-layer transmission modes such as modulation scheme, FEC rate, transmission power, or a combination thereof, is selected in accordance with TCP dynamics and wireless channel conditions. The simulation results, which performed in IEEE802.11a network, show a significant enhancements in the TCP throughput.

2.4.3 TCP and Resource Allocation

A thorough overview of cross-layer design for resource allocation algorithms in the third generation of wireless networks is given by [56], where TCP over CDMA is also addressed. In this paper, authors suggest that obtaining a set of link-layer design parameter from the cross-layer interaction with TCP, can improve the throughput of TCP.

TCP-aware resource allocation algorithms over a CDMA network are studied in [57], the objective being to maximise throughput. This scheme, adapts the channel rate in response to the TCP sending rate, allowing it to trade off channel rate and the error rate in a TCP-friendly manner. Using an analytical model, an improvement of 15 to 20 percent is reported in the throughput of TCP.

A joint congestion control and power allocation in a CDMA based wireless network is proposed in [58], which increases the end-to-end throughput and energy efficiency of multiple transmissions in the multihop wireless networks. The author expands the joint problem to a more generalised network utility maximisation framework, and show that this framework can achieve the optimal balance between the transport and physical layer in multihop networks.

In the context of IEEE 802.16, reference [59] proposes a TCP-aware resource allocation scheme which estimates the bandwidth demand based on the long-term data rate, and allocates resources accordingly with no explicit information from the sender.

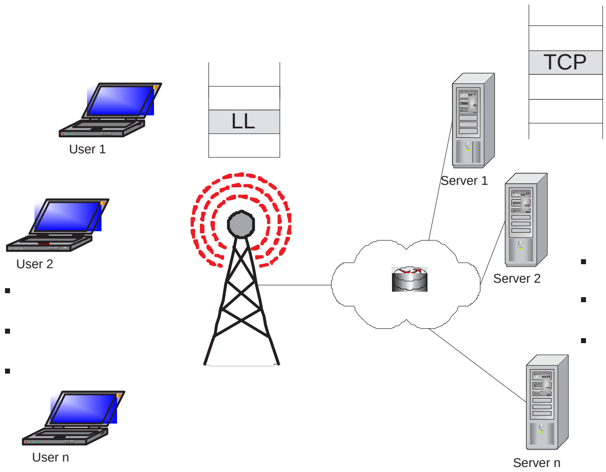


Figure 2.2: Cross-Layer Interactions between TCP at the end-host and the Link-Layer at the wireless base station.

2.4.4 Cross-Layer Interactions between TCP and Link-Layer

In the thesis, the cross-layer approach is a top-down cross-layer interaction between TCP and link-layer. Therefore, the information of TCP state diagram such as cwnd, TCP flavour, and the timing information such as RTT are required at the wireless link-layer. The methodologies to transfer these information from the end-host TCP to the link-layer at the wireless base station are discussed in the followings. The structure of this cross-layer interaction is depicted in Figure 2.2.

Various methods are presented in the literature to estimate RTT either actively or passively at any router in the middle of the end-to-end path. For active measurement, TCP timing information can be included in the Timestamp option of the TCP header. The Timestamp option allows the sender to place a timestamp value in any transmitting packet, thus the receiver reflects this value in the ACK and the sender can calculate the RTT accordingly [60]. Similar calculation can be done in the middle routers such as wireless Access Point (AP).

On the other hand, TCP header may be encrypted and can not be read at the middle routers. Therefore, other techniques are presented to measure the RTT passively that are also precise in fairly high probability. For example, proposed technique in [61], monitor data stream to detect cyclical patterns caused by TCP's self-clocking mechanism. The key idea is based on the TCP self-clocking that results in the repeating pattern in packet arrival for each TCP connection. These algorithms employ autocorrelation to find the period of the packet arrival pattern, which is the RTT. The passive measurement based on the three-way handshake is presented in [62], and show that 90% of the passive measurements are within 10% of the precise RTT value.

TCP flavours can be identified via the mechanism presented in [63]. Here, the TCP flavour and state are determined by monitoring changes in the cwnd, where the cwnd can be estimated passively at any router within the network. This technique can be implemented at the wireless AP, and identify the cwnd of TCP flow, and also its utilised flavour at the end-host.

Chapter 3

TCP-aware Link-Level FEC to Improve End-to-End Performance

Among the tapestry of solutions that have made the development and success of the Internet possible, the TCP, and numerous variations thereof, have become the norm for the provision of a reliable end-to-end transport layer. As the take-up of wireless technologies further proliferates, it is expected, for compatibility reasons, that TCP will be used much more often over wireless networks. A significant amount of research has therefore evolved over the past few years, aimed at improving the performance of TCP over wireless networks [40]. The mass of prior art on designs to improve TCP over wireless links has thus far assumed that all competing flows are of the same flavour. However, with the spiralling number of TCP flavours in existence over the Internet [64], the end-to-end performance characteristics of TCP, particularly in cases of random—wireless—packet loss, have become increasingly diverse.

This research studies a more generalised framework, whereby the different competing TCP flows are heterogeneous in nature, i.e., they can be based on different TCP flavours—such as for example Reno [9], NewReno [10], Westwood [16], or using the SACK option [12]. In this context, because TCP flavours react differently to random packet losses, a significant degree of unfairness among the competing flows can surface in terms of achieved end-to-end throughput. The performances seen by the different TCP flows depends not only on packet loss rates over the wireless link, but also on the exact combination of currently utilised TCP flavours over that wireless link. Hence, designs to improve end-to-end TCP performance should also take into consideration the mix of TCP flavours operating over the wireless link [65; 66].

To this end, based on the constraints imposed by the wireless link, an optimisation problem is detailed that strives to maximise Jain's fairness index [67] in respect to the throughputs achieved among the competing TCP flows. Moreover, fairness among the different TCP flavours competing over wireless medium is proposed to improve, through introducing an algorithm that dynamically enhances the link-layer FEC code rate [68; 69].

The investigated scenario in this chapter assumed the co-existence of various TCP flavours competing over the wireless link. The network level simulator OPNET that provides real TCP implementations is used. Thus, the simulation scenarios also investigate the behaviour of real TCP implementations using the proposed scheme. In the above mentioned scenario following issues are addressed,

1. A novel analytical framework is proposed to maximise fairness among TCP flows by adapting the PER that can be seen by each corresponding TCP flow.

2. In order to adapt the PER of TCP, the link-level FEC coding rate is selected dynamically according to the results of the proposed optimisation problem.

3. To solve the proposed problem at the wireless base station in real-time, a heuristic approach is presented, where the Logarithmic Barrier method is used. The convergence of this heuristic approach is also investigated.

4. Through extensive simulation studies using OPNET modeler, it has been shown that this novel cross-layer scheme considerably improves the fairness over wireless links among flows comprised of different TCP flavours. Further

investigations are made on the effect of the proposed scheme on the aggregate achieved throughput of the competing heterogeneous TCP flows.

Despite the fact that adapting the FEC rate with respect to the TCP performance has been studied in the literature [52; 70], issues regarding the existence of various competing TCP flavours have not been previously reported.

The remainder of this chapter is organised as follows. In Section 3.1, the literature in terms of reliability at the link-layer that provided by FEC scheme, and its adaptation to enhance the performance of TCP is studied. The research works involving fairness among TCP flows over wireless networks are also reviewed in this section. After discussing the effect of TCP flavour on the throughput of TCP, Section 3.2 verifies the utilised analytical expressions against real TCP implementations in the simulation platform. In Section 3.3, Jain's fairness index is used as an objective function for the proposed optimisation framework, and the mechanism to achieve the resulted PER by the appropriate selection of FEC rate is discussed. Section 3.4 presents some of the numerical results, and in order to solve the proposed problem in real-time, a heuristic approach is presented. Section 3.5 investigates the performance of the proposed scheme within the framework of a network simulation platform. Finally, this chapter concludes in Section 3.6.

3.1 Background Study

In this section the state of the art in the two related topics are addressed. Firstly, the related literature which investigates the effect of link-level FEC on the performance of TCP is discussed. Thus, some proposals for adapting the FEC coding rate in order to improve the performance of TCP are presented. Secondly, the related literature that addresses the issue of fairness among TCP flows over wireless networks is studied. It is worthwhile mentioning that none of the above mentioned research areas have considered the co-existence of various TCP flavours. Another issue which needs to be addressed in this section is the methods of transferring cross-layer information from TCP at the end-host to the link-layer at the wireless base station, which is fully discussed in Section 2.4.4.

3.1.1 FEC Adaptation to Improve TCP Performance

In wireless networks, in order to counter the effects of random losses, some degree of additional reliability is often provided at the link-layer. This reliability is usually achieved through link-level ARQ in conjunction with FEC. The design issues of FEC are addressed here, while the issues related to ARQ are discussed in Chapter 4.

FEC at the link-layer can improves reliability by creating redundant data from the original data sequence, prior to transmission. This redundant data allows the receiver to correct a proportion of errors caused by corruption from the wireless channel. It should be noted that when a channel has a very low signal to noise ratio, adding the redundancy, can even decrease the performance of the wireless link ([71], Chapter 8.2). In other circumstances, increasing the redundancy improves the reliability of FEC that also increases the channel load. This tradeoff is studied in [52] where the achieved TCP performance with respect to the FEC rate has been shown and the gains that can be attained in the end-to-end TCP throughput has been studied.

Moreover, in [70] the optimal design of a hybrid FEC/ARQ scheme with respect to the TCP throughput is presented. Although both TCP Reno and TCP NewReno versions are studied in [70], the co-existence of these two flavours is not investigated. In fact, this research work investigates a single TCP flow scenario. Therefore, it can be seen that the mass of previous work on the FEC rate selection with respect to the TCP performance, have neither considered the co-existence of various flavours nor the effect of such a combination on the FEC rate selection.

3.1.2 Fairness Among TCP Flows in Wireless Networks

Since the majority of applications on the Internet use TCP, TCP's fairness has been well studied in the literature. The unfairness among TCP flows in WLAN is investigated in [72; 73], where it has been shown that the base station's buffer size affects fairness. Reference [72] takes into account that the majority of applications involve download rather than upload, and proposes a rate control mechanism that modifies the TCP advertised window size in order to avoid loss in the downlink buffer. In [73], a smoother rate control mechanism is proposed to improve fairness in two highly congested scenarios that may cause starvation to TCP connections. The first scenario studied is the case where packets that belong to multiple TCP flows are competing in the WLAN base station transmission buffer, and the second scenario is where the base station is congested with TCP ACKs to be transmitted to the mobile users. In that paper, authors evaluate the coefficient of variation of throughput as the "unfairness index", thereby demonstrating that the scheme's performance is not affected by the version of TCP—the Reno and NewReno versions are studied. However, the effect of combining multiple TCP versions on fairness degradation is not studied.

3.1.3 Measure of Fairness: Jain's Fairness Index

Various fairness measures have been proposed in the literature. The Jain's fairness Index, which was conceived to measure fairness in computer networks [67], is a very well used measure of fairness due to its advantageous mathematical properties. The Jain's index is independent of the scale of the allocation metric, and is bounded in the range [0, 1]. In addition, Jain's index is continuous such that any change in allocation also changes the fairness. This index is widely used in wired and wireless networks [74] as a quantitative measure of fairness. In this work, Jain's index is therefore utilised to measure fairness among TCP flows over the wireless network.

3.2 Throughput Comparison of Various TCP Flavours

In the presence of losses, the behaviour of TCP congestion control varies depending on the TCP flavour. This research work is concentrated on the three TCP flavours: TCP Reno, and TCP NewReno, which are very well-used over the Internet, and TCP Westwood, which is built on TCP Reno. Furthermore, the presence of the SACK option in TCP Reno is studied.

The three mentioned TCP flavours are analytically modelled in the literature [26; 27; 29; 30]. The analytical expression of the SACK option is also presented in [28]. These models express the TCP throughput as a function of the PER and the end-to-end RTT that each TCP flow experiences. Using these models, the effect of PER on the TCP throughput of each flavour is studied. Furthermore, each TCP flavour is also simulated in a single-flow scenario in OPNET, and the results of the simulation and analysis are compared.

To achieve this, the simulation code for TCP Westwood in OPNET has been created for the research of this thesis based on adopting the model which was already available in ns-2. The detail of the OPNET modeler, and TCP Westwood implementation is presented in Appendix B. The same conditions of 100 ms RTT, 6 Mbps bottleneck link, and a random packet loss rate varied from 10^{-5} to 10^{-2} are applied to the analytical model and the OPNET simulations; moreover, the simulations are all performed over a download file size of 16 MB—approximately $11,000$ packets—where packet size is 1460 B. Figure 3.1 clearly shows that these four versions of TCP react significantly different when the packet loss increases. In addition, Figure 3.1 compares the OPNET TCP models, with the analytical models.

In the rest of this chapter, it is assumed that packet loss is detected only via DupACKs, such that TCP does not face timer expiration. Timer expiration occurs mainly from the loss of an ACK packet, which can be attributed to the use of cumulative acknowledgement. Thus, the probability of loosing an ACK packet is low—much lower than for data packets—even in wireless channels with high error

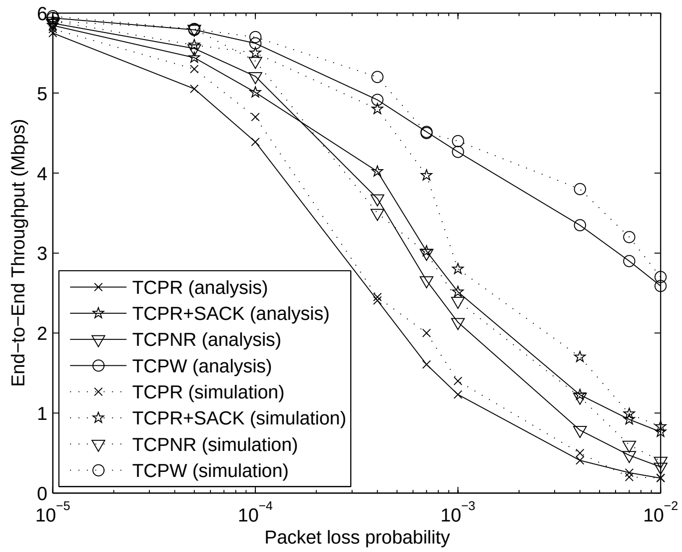


Figure 3.1: Single-flow analytical model and simulation results comparisons for TCP Reno, TCP Reno+SACK, TCP NewReno, and TCP Westwood

probability. Therefore, generality is not affected by this assumption. TCP throughput models assuming packet loss is only detected via DupACKs are detailed in Section 2.1.4. Therefore, for the rest of this chapter, Equations (2.4), (2.6), (2.7), and (2.13) are used as the throughput expressions of Reno, Reno+SACK, NewReno, and Westwood respectively.

3.3 Fairness among TCP flows in Multiple-Flavour Scenarios

Where all flows are based on the same TCP flavour, congestion control algorithms guarantee fairness among TCP flows. However, in cases where multiple TCP flavours are coexisting over the lossy wireless link, fairness is affected by different reactions of TCP flavours to packet losses. In this work, the well-used fairness index, Jain's index [67], is used as a notation of fairness that is defined by the following equation:

$$J(\mathbf{E}) = \frac{\left(\sum_{i=1}^{n} x(e_i)\right)^2}{n \cdot \sum_{i=1}^{n} x(e_i)^2}, \tag{3.1}$$

where n is the number of TCP flows, $[\mathbf{E}]_i = e_i$ is the associated PER for flow i, and $x(e_i)$ expresses the ratio between the TCP throughput and the optimal throughput that can be achieved by each TCP flow i.e., $x(e_i) = B_k\left(e_i\right)/B_{k_{Optimal}}$. The optimal TCP rate is defined as the throughput of flow i when all the other $n-1$ flows have similar TCP flavour to flow i.

3.3.1 Problem Definition

The n TCP flows are considered, denoted by $i = 1..n$, where each TCP flow i can be served by any of the four TCP flavours, i.e. Reno, Reno+SACK, NewReno or Westwood, enumerated by $k = 1..4$. These n flows compete for the limited capacity of the wireless link, W, and their throughputs are affected by the wireless error rate as described in Section 3.2. Therefore, the throughput of each flow, $B_k\left(e_i\right)$, is given as a function of the corresponding flow's PER, e_i. It is further assumed that the probability of a packet being in error can be adjusted according to $e_i \cdot 10^{\pm\xi}$ due to the specified link-layer error recovery algorithm—FEC code rate, C_i, in this case. Therefore, the description of TCP throughput can be adjusted to $B_k\left(e_i \cdot 10^{\pm\xi}\right)$. The aim is to maximise fairness among the n flows that use different TCP flavours. As mentioned earlier, $x(e_i)$ is the normalised TCP throughput using the optimal throughput value that can be achieved by each flow. The optimal TCP rate is calculated for the minimum feasible value of PER, $e_i \cdot 10^{-\xi}$. Under the above assumptions, an optimisation problem is formulated with the objective function being to maximise the Jain's fairness index. In addition, it has been illustrated that maximising the fairness index under this framework does not affect the overall TCP throughput significantly. The proposed non-linear optimisation problem is outlined below:

$$(\mathrm{P}): \quad \text{maximise } J(\mathbf{E}) = \frac{\left(\sum_{i=1}^{n} \frac{B_k(e_i)}{B_{k_{Optimal}}}\right)^2}{n \cdot \sum_{i=1}^{n} \left(\frac{B_k(e_i)}{B_{k_{Optimal}}}\right)^2},$$

subject to,

$$\sum_{i=1}^{n} B_k\left(e_i\right) \leq W, \quad \forall k \in \{1, ..., 4\} \tag{3.2}$$

$$e_i \cdot 10^{-\xi} \leq e_i \leq e_i \cdot 10^{\xi}, \quad \forall i \in \{1, ..., n\} \tag{3.3}$$

$$0 \leq e_i \leq 1. \quad \forall i \in \{1, ..., n\} \tag{3.4}$$

Although PER of e_i is a discrete variable, assuming that TCP packet size can be changed accordingly, e_i can be converted to a continuous variable. The TCP throughput is a non-linear but differentiable function over $[e_i \cdot 10^{-\xi}, e_i \cdot 10^{\xi}]$. The details of the congestion control algorithm behaviour for TCP Reno, TCP NewReno and TCP Westwood, are described in Section 2.1.2. Moreover, the throughput expressions of Equations (2.4), (2.6), (2.7), and (2.13) are replaced B_k in this formulation. It is also assumed that TCP connections are sufficiently long-lived.

3.3.2 Link-Layer FEC Code Rate Selection

It is assumed that TCP packets' size are MSS Bytes, which are fragmented into link-layer frames of m Bytes and later coded into blocks of M Bytes to transmit over the wireless link. A convolutional coding is assumed with the code rate equal to C where $C = m/M$. For the convolutional coding, Hamming distance is defined as the distance between two code words, in which the minimum distance of the code is called minimum free distance and denoted by d_{free} ([71], Chapter 8.2). The block error rate, l, over the block size of M is defined as a function of d_{free} which also depends on the C and the Bit Error Rate (BER) of the wireless link, b, such that, $l = f\left(C, b\right)$. Given the fixed value of b—at the same condition over wireless channel—and having different values for C, the value of l and consequently e vary.

In other words, increasing the redundancy results in decreasing the packet error rate e and vise versa.

Given different code rates of C, the values of function f is calculated for the wireless channel BER in the range $[10^{-6}, 0.1]$. Thus, Figure 3.5 shows the wireless channel BER, b, versus the PER of TCP, e, for various values of C. The specifications of the utilised encoder/decoder and further discussion on Figure 3.5 are given in Section 3.5.2. Knowing the current BER of the channel, the optimal code rate for the link-level FEC can be found by moving vertically on the curve of Figure 3.5 from the initial PER of the corresponding TCP flow to its desired PER that is the result of solving problem (P). Assuming the initial PER for flow i is e_i, the desired PER is in the range $[e_i \cdot 10^{-\xi}, e_i \cdot 10^{\xi}]$, thus the feasible PERs shown on the y-axis should be within this range.

3.4 Observations Based on Numerical Investigations

To observe the performance of such a scheme, the optimisation problem (P) has been solved using MATLAB's optimisation Toolbox. As an illustrating example 15 TCP flows are assumed with PER as a random variable in the range $[10^{-5}, 10^{-2}]$, and ξ is 2, thus the exponent of PER is bounded by ± 2. It is assumed that the bottleneck is wireless link, thus the PER is only affected by the loss at the wireless link. The wireless channel is symmetric, therefore the received ACK and sent data face similar channel conditions. Given the same termination points for all the flows in the wired network, the RTT for all flows are equal—100 ms in this section. Thus unfairness arises solely from the reactions of TCP flavours to packet losses.

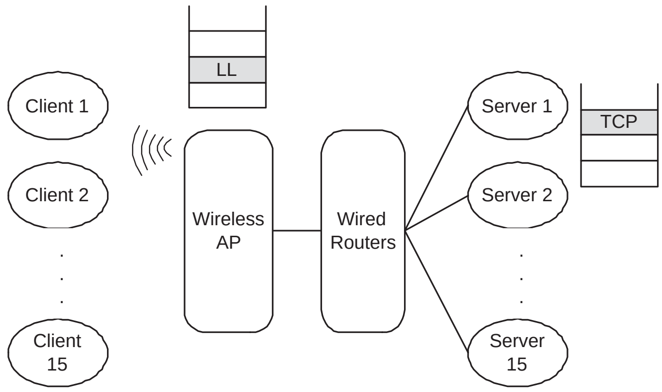
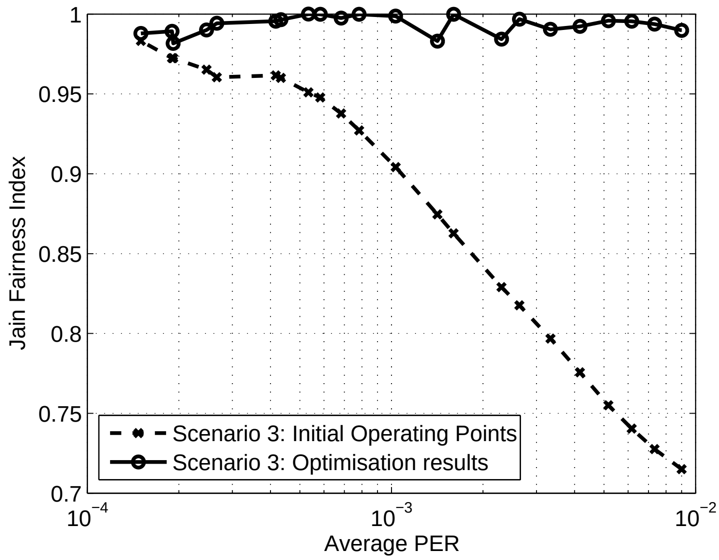


Figure 3.2: The investigated network architecture.

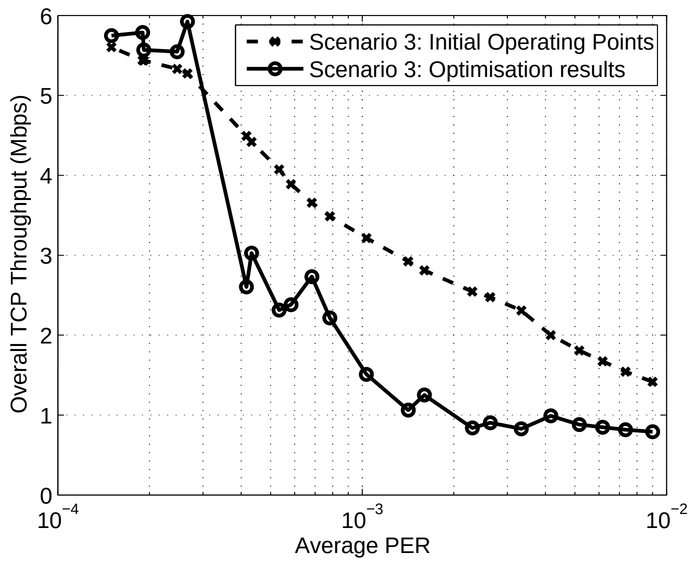
3.4.1 Numerical Results

In order to strive for generality, three combinations of TCP flavours are studied. The first scenario consists of four TCP Reno, four TCP Reno+SACK, four TCP NewReno, and three TCP Westwood. In the second scenario, there are three TCP Reno, five TCP Reno+SACK, five TCP NewReno, and two TCP Westwood. Finally, the combination of TCP flavours in the third scenario is three TCP Reno, three TCP Reno+SACK, three TCP NewReno, and six TCP Westwood. All the three scenarios are simulated over a network configuration as shown in Figure 3.2, where all the traffic passes through a single wireless AP.

The results of these three scenarios are summarised in Table 3.1, in which the increase of up to 46% in fairness index can be seen. The plotted results in Figure 3.3 are from the third simulation scenario, in which the achieved fairness among TCP flows as well as the overall throughput are presented. Observe from this figure, it can be seen that improvement in the fairness index is more significant in high values of PER. Similar observation can be seen from the aggregated throughput that is slightly decreased when the probability of packets in error gets higher. On average, over the range of PER values and combination of TCP flavours, fairness is increased approximately 12%.



(a) Fairness Index



(b) Aggregated TCP throughput

Figure 3.3: Numerical observations from the TCP-aware FEC scheme compared with what would otherwise be achieved in scenario 3.

Table 3.1: Fairness improvement as achieved by the proposed TCP-aware FEC scheme.

	Average Increase	Maximum Increase
Scenario one	13%	46%
Scenario two	10%	43%
Scenario three	12%	38%

3.4.2 Using the Logarithmic Barrier Method to Solve the Problem in Real-Time

In order to proceed further, a method to solve the proposed problem in real-time is required, thus the optimisation can be performed at the wireless base stations. In this respect, the optimisation problem is transformed such that Newton's method can be applied ([75], Chapter 9.5). Therefore, the constrained problem (P) is approximated to an unconstrained optimisation problem with the Logarithmic barrier function ([75], Chapter 11.2). However, through approximating with the Logarithmic barrier function, the inequality constraints of Equations (3.2)-(3.4) can be implicit in the objective function—see problem P. The optimisation problem can therefore be rewritten as follows,

$$(\mathrm{P}'): \quad \text{minimise} \;\; Y(\mathbf{E}) = -J(\mathbf{E}) - 1/t \cdot \Phi(\mathbf{E}), \tag{3.5}$$

where

$$\begin{aligned} \Phi(\mathbf{E}) = \;& \log\left(-\sum_{i=1}^{n} B_k(e_i) + W\right) + \\ & \sum_{i=1}^{n} \log\left(-B_k(e_i) + B_k(p_i \cdot 10^{-\xi})\right) + \\ & \sum_{i=1}^{n} \log\left(B_k(e_i) - B_k(e_i \cdot 10^{\xi})\right). \end{aligned} \tag{3.6}$$

The parameter t sets the accuracy of the approximation; in the other words, the approximation becomes more precise as the parameter t increases. Unfortunately, calculations become less tractable as t gets larger. The modified objective function is convex, therefore Newton's method can be used to solve it–for more details on Newton's method see Appendix A.

The Hessian Matrix is defined, H, to formulate the Newton step, as follows,

$$H = -\nabla^2 J(\mathbf{E}) + (-1/t) \cdot \nabla^2 \Phi(\mathbf{E}). \tag{3.7}$$

Algorithm 1 Newton iterations to solve problem P

1) Start from the current operational point as $e_{k=1}$.
2) Compute the first gradient, $\nabla Y(\mathbf{E})$, and the second gradient, H, of the objective function.
3) Compute Δe_{nt} $(= e_{k+1} - e_k)$ from Equation (3.8).
4) If $\|\nabla Y(\mathbf{E})\| \leq \varepsilon$ stop,
else go back to 2.

the Newton step, Δp_{nt}, can be defined as,

$$\Delta p_{nt} = H^{-1} \cdot (\nabla(-J(\mathbf{E})) + (-1/t) \cdot \nabla\Phi(\mathbf{E})) \,. \tag{3.8}$$

where ∇ and ∇^2 are the first and the second gradient.

The optimal values of e_i are the results of the Newton iterations initiated from the current operation point of each flow. This iteration is described in further detail in Algorithm 1, where ε is a relatively small value.

In order to find the correct value of parameter t for this approximation, it is increased in sequential steps. On the other hand, when t is large, the problem is difficult to be solved by Newton's method, as its Hessian varies rapidly near the boundaries. In this problem, setting t to five can lead to the optimal region. The concept of Logarithmic barrier method is detailed in Appendix A, where a simple example is also solved using Newton method.

Another crucial issue that needs to be examined here, is the convergence of this solution. It can be seen in Figure 3.4 that, given this approximation, the problem is solved in a small number of iterations. Figure 3.4 shows that, the heuristic approximation is solved in a maximum of 50 iterations, while the average number of required iterations is approximately 12. Moreover, in 70% of cases, the optimal value is attained in less than 15 iterations.

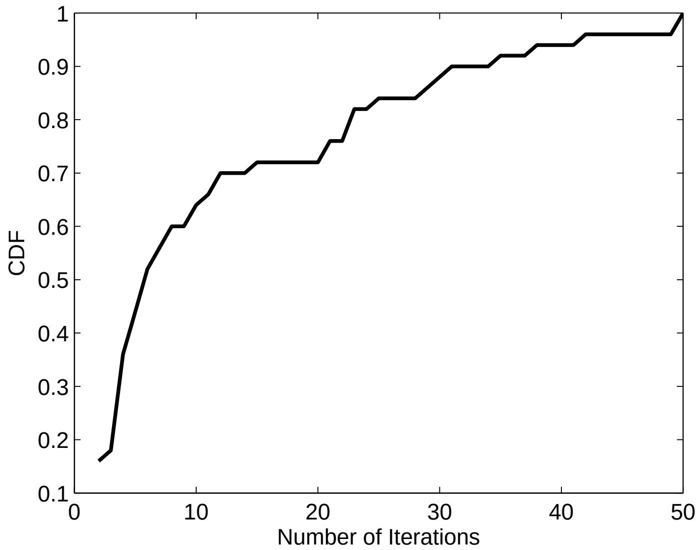


Figure 3.4: CDF of the number of iterations required in solving the optimisation problem using Newton's method

3.5 Performance Investigations

To further investigate the performance of the proposed scheme and its effect on real TCP implementations, the OPNET simulation platform is used. The implementation of TCP in OPNET is based on its RFCs. TCP Reno and TCP NewReno as well as the SACK option were all already supported within OPNET. However, TCP Westwood is implemented for the purpose of this research work within the OPNET modeler based on the available model from Network Simulator 2 [76]. The model has been verified, through comparison with published simulation results and by comparison with an analytical model, as well as by liaising with the designers of TCP Westwood. More detail on the OPNET modeler and the TCP Westwood model are presented in Appendix B.

3.5.1 Simulation Parameters

In these simulated scenarios, wireless clients set up connections with wired servers via a single wireless AP and wired routers—see Figure 3.2. Each wireless client connects to a unique server, where the bottleneck is assumed to be at the wireless link. The wireless channel is modelled with an ITU indoor path loss model, and Lognormal shadowing with the standard deviation equal to 4 dB. Given d the users' distance from the AP in meters and f_c the operating frequency in MHz, the path loss can be written as,

$$PL = 30\log(d) + 20\log(f_c) - 12. \tag{3.9}$$

In addition, low mobility users are assumed, and channel parameters are constant over each two-second interval. This is because, every two seconds, information on the channel error rate is updated at the AP, whereby the optimisation problem is then triggered to renew the corresponding code rate of each flow. In different simulation scenarios, the distances of wireless clients from the AP are changed within the range of 20 m to 70 m, and RTTs are varied between 20 ms and 300 ms. Mobile users are distributed uniformly over the cell with radius R. To uniformly distribute the mobile users, the angle θ is chosen uniformly and the radius is $r = R \cdot \sqrt{z}$ where z is generated uniformly in [0, 1] [77]. Other specific simulation characteristics are detailed in Table 3.2.

Although the proposed scheme can be applied to different wireless networks, the concentration in this work is on 802.11a, as the properties associated with OFDM has led to its consideration as a candidate for new generation of wireless networks.

3.5.2 Convolutional Encoder/Decoder

In this section, the encoder/decoder utilised to attain the simulation results is discussed in more detail. The NASA standard convolutional encoder/decoder is assumed, which is well implemented e.g. in Actel encoder/decoder core [79], and can support selectable code rates of $\frac{1}{3}$, $\frac{1}{2}$, $\frac{2}{3}$, $\frac{4}{5}$, $\frac{5}{6}$, $\frac{7}{8}$. The constraint length K is equal to 7 and the polynomials are $g_0 = 171$, $g_1 = 133$. Values of the minimum distance, d_{free}, for the described encoder/decoder are calculated from the convolutional Trel-

Table 3.2: Simulation Parameters for the TCP-aware FEC scheme

Simulation duration	120 s
FTP servers	16 MB file download size
HTTP servers	HTTP1.1 webpage interarrival time (s): Lognormal (μ = 0.22, σ = 2.64) HTML object size (KB): Lognormal (μ = 7.90, σ = 1.76) Number of images per webpage: Gamma (k = 0.14, θ = 40.32) Image size (KB): (Lognormal μ = 7.51, σ = 2.17) where μ and σ are from the mean and standard deviation of the Normal distribution, and k and θ are the shape and scale parameters of Gamma distribution [78].
TCP MSS	1460 B
MAC packet size	365 B
MAC specification	IEEE 802.11
PHY characteristic	OFDM (IEEE 802.11a)
Operating Frequency	f_c = 5.4 GHz
Data Rate	6 Mbps

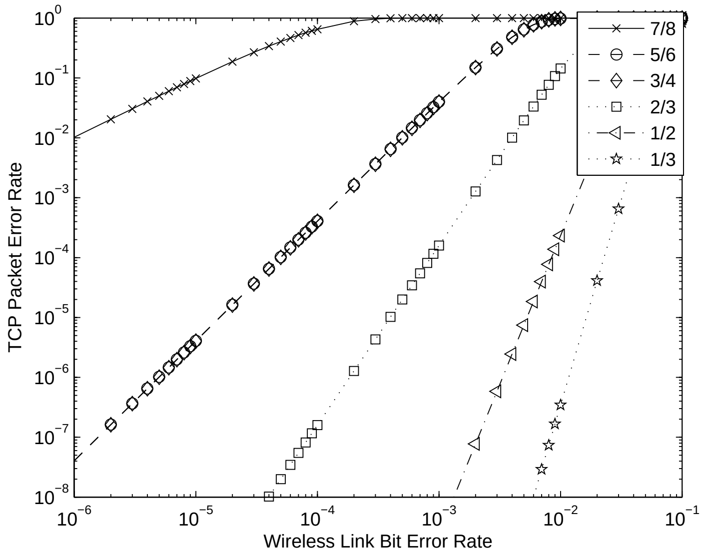


Figure 3.5: TCP packet loss versus link BER in various Convolutional code rates with K = 7, g_0 = 171 and g_1 = 133.

lis diagram with $K = 7$ that are equal to 15, 10, 6, 5, 4 and 3 respectively for the six mentioned code rates [79]. Given MSS the Ethernet value of 1460 B and m, 365 B, the values of function $f(C, b)$ can be derived numerically. Figure 3.5 shows the TCP PER, e, versus wireless link BER, b, for the various code rates. As discussed earlier, knowing the BER of the wireless channel, the link-level FEC code rate for the corresponding TCP flow can be found by moving vertically on the curve of Figure 3.5 from the initial value of PER to its desired value that is the result of solving problem (P).

3.5.3 Simulation Scenarios and Simulation Results

The existing 15 wireless clients are connected to 15 servers—as indicated in Figure 3.2. The combination of TCP flavours are 4 TCP Reno, 4 TCP NewReno, 4 TCP Westwood, and 3 TCP Reno with enabled SACK option. The cell is assumed to have a circular shape with the wireless AP in the centre. In scenarios one, and three users are randomly distributed in the cell using the uniform distribution discussed earlier. In scenario two, distances of users from the wireless AP refers to the random location

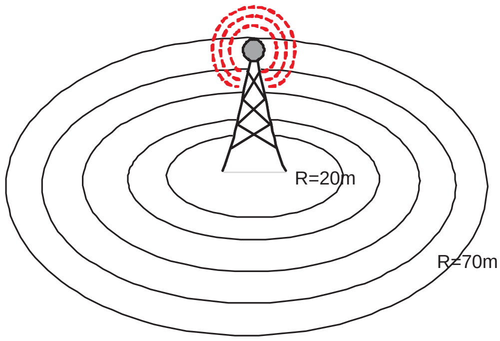


Figure 3.6: Users' distribution around the wireless AP while keeping fixed distances from the AP.

of a user on a circle around the AP—as plotted in Figure 3.6—with the radius R. Hence, in this scenario all users keep the fix distance from the AP in each simulation run. Locating mobile users on these circles with the constant distances from the AP in each simulation run, keeps the path loss coefficient constant, thus examining the random slow fading effect. On the other hand, changing the distances from the AP—moving mobile users to the next circle—includes the effect of path loss which is the deterministic part of the channel loss. Every presented simulation scenario is performed five times and the results are the average of these five simulations.

The benchmark is a system that simply adapts its FEC code rate based on the channel quality. This code rate is assumed to be taken from one of the six available rates—see Figure 3.5—based on the channel BER, and is updated once every two seconds. The results of the TCP-aware optimisation scheme are compared with this benchmark. The results in terms of Fairness Index versus RTT, and mobile users' distances to the AP are presented and compared with the benchmark problem. Aggregated end-to-end throughput is also studied in the scenarios detailed in the sequel.

3.5.3.1 Simulation Scenario One: Various end-to-end RTT values

The first simulated scenario studies the achieved fairness by the proposed scheme in the presence of different end-to-end path RTT values. In the subsequence runs of this simulation scenario, the RTTs to the 15 FTP servers in the wired part of the

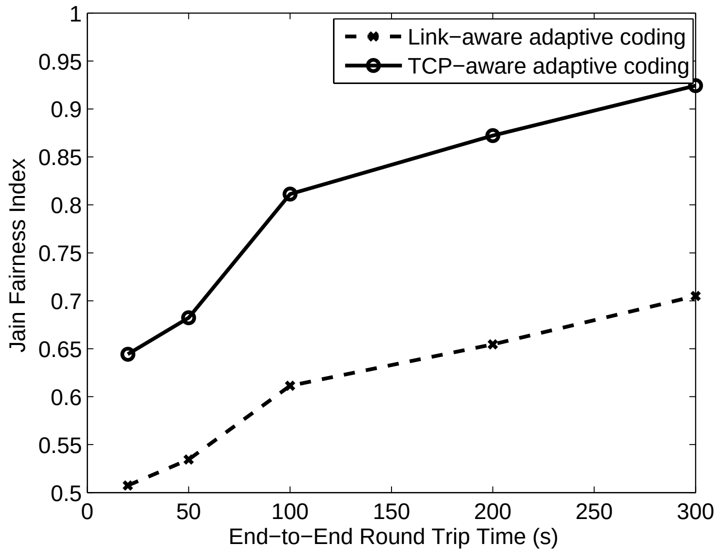


Figure 3.7: Scenario one: Jain's index versus end-to-end RTT (ms).

network, are varied from 20 ms to 300 ms. Moreover, users are uniformly distributed within the cell with radius $R = 60$ m—the uniform distribution of users in the cell is with respect to the earlier described method. In Figure 3.7 that presents the achieved fairness index versus RTT, the average improvement of 30% in fairness index can be seen.

3.5.3.2 Simulation Scenario Two: Changing the users' distance from the AP

In scenario two, the effect of moving mobile users further/closer from/to the wireless AP on the achieved fairness is investigated. In this respect, in each run of simulation scenario two, all 15 mobile users are located on a specific circle—as indicated in Figure 3.6—that results in changing the users' distance from the AP in each run from 20 m to 70 m. The same set of uniformly distributed random RTT values are in place for all of the simulation runs of scenario two, which are in the range [20 ms, 300 ms]. Moreover, the 15 servers are FTP servers. The achieved fairness in this scenario is plotted in Figure 3.8, where an average of 30% improvement can be observed.

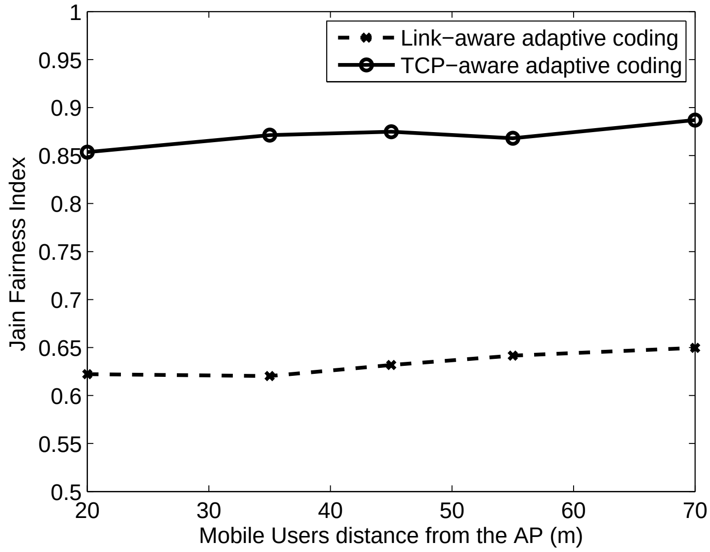


Figure 3.8: Scenario two: Jain's index versus users' distance from the AP (m).

Additional observation from this figure reveals that, first, the level of achieved fairness is not affected by moving all the 15 mobile users further/closer from/to the wireless AP—unlike the observation from Figure 3.7. Second, having diversity among the end-to-end RTTs in this scenario, which is caused by assigning random RTTs to each TCP flow, results in the high level of achieved fairness using the scheme presented in this research work—over 85% fairness for every run of the simulations in Figure 3.8.

In addition to the achieved fairness among competing TCP flows, the overall end-to-end throughput using the proposed scheme is examined. The aggregated end-to-end throughput is plotted versus the users' distance from the AP in Figure 3.9. The results present in this figure show that the overall TCP throughput is slightly increased when users are close to the AP and slightly decreased as users get further from the AP.

3.5.3.3 Simulation Scenario Three: Different traffic types

Scenario three investigates the performance of the proposed scheme under different traffic types: FTP and HTTP traffic. In the two runs of this scenario the RTT

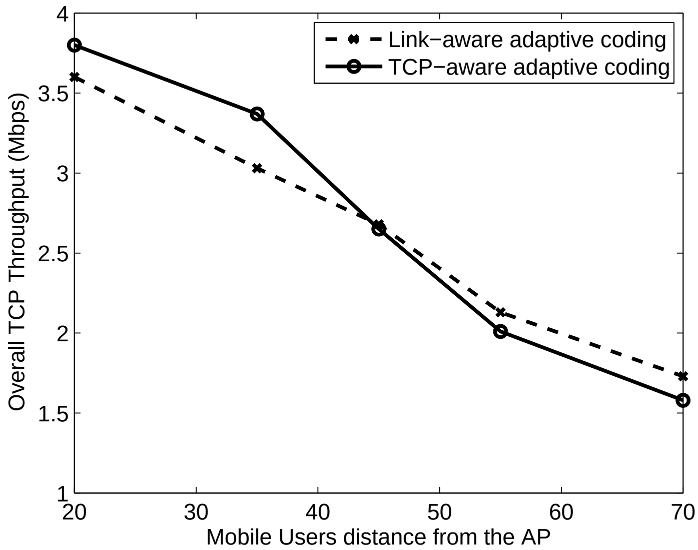


Figure 3.9: Scenario two: Overall end-to-end throughput versus users' distance from the AP (m)

values are random values uniformly distributed in the range [20 ms, 300 ms], and users are uniformly distributed within the cell with radius $R = 60$ m. The first run of scenario three is performed with 15 FTP servers, and the second run is performed with 15 HTTP servers with the properties described in Table 3.2. The results of this scenario are presented in Table 3.3, in which the average achieved fairness Index is detailed. It can be seen that, the improvement in the fairness index in both the traffic types is significant, and approximately 30%. It is also clear that fairness index is not affected by different traffic types.

3.5.3.4 Overall Results

The results presented in Figures 3.7-3.9 and Table 3.3 show not only that the fairness index is improved using the proposed scheme in this chapter—by approximately 30% in average—but also that the overall TCP throughput is minimally affected. More generally, the proposed optimisation framework yields improvements in fairness in a variety of conditions regarding end-to-end RTTs, users' distances from the AP, and also combinations of TCP flavours, and traffic types. The improvement in fairness is significant in both the analysis and the OPNET simulation scenarios, while at the

Table 3.3: Jain's Index in scenarios three using FTP and HTTP applications

	Jain's Fairness Index	
Traffic Type	FTP	HTTP
link-aware adaptive coding	0.615	0.653
TCP-aware adaptive coding	0.832	0.875

same time the effect on the end-to-end performance is minimal.

3.6 Concluding Remarks

In this chapter, a cross-layer mechanism is presented to dynamically set the FEC rate at the link-layer on a per-flow basis, given the end-to-end TCP flavour of that flow as detected at the wireless link. Utilising information on the TCP flavour of each flow, a framework is detailed to improve fairness among heterogeneous TCP flows that compete over a shared wireless link. To allow real-time implementation, an heuristic approach is presented, in order to ascertain the optimal link-layer coding rate that maximises fairness at the wireless base station. The convergence properties of this proposed heuristic approach have been studied.

The analysis indicates that using the proposed scheme, Jain's fairness index can be improved significantly; moreover, the overall TCP throughput is minimally affected. Comparative simulation studies using OPNET modeler have also been performed, where the utilised implementations of different TCP flavours precisely match their respective RFCs. These simulations, under various packet loss probabilities and RTT conditions, show an improvement in fairness of approximately 30%.

Chapter 4

TCP-aware ARQ Mechanism to Improve End-to-End Performance

Among other functionalities, TCP provides reliable data transmission that is obtained through positive acknowledgement. In the event of loss, TCP performs retransmissions, thus reducing the overall TCP throughput. Study on the impact of retransmissions on the TCP throughput resulted in the improvements that are embodied as Fast Retransmit and Recovery in TCP algorithms [9]. More recently, extensive efforts have been expended on modifications to improve the performance of TCP via advanced algorithms implemented in other layers of the protocol stack. Such efforts usually attempt to avoid the TCP congestion control mechanism being incorrectly triggered as a result of random wireless errors. In addition, the traditional and widely implemented approach to increase the reliability of the wireless link is based on the usage of an ARQ protocol at the link-layer.

Retransmissions at the link-layer has two main advantages compared with TCP end-to-end retransmissions. Firstly, retransmissions at the link-layer are performed faster, due to the local link having a much smaller RTT—this phenomena can be seen in Figure 4.1. Secondly, successful retransmissions at the link-layer can hide the packet losses from TCP, thus the cwnd will not be incorrectly altered. It is worth mentioning that the incorrect changes in cwnd due to the wireless error can cause the wireless link to be under utilised. On the other hand, scalability and low computational complexity, results in the employment of the ARQ in most of the wireless link-layers.

The stop-and-wait ARQ protocol, which is used in this work, operates based on the positive ACK. Unacknowledged packets are therefore retransmitted if the required number of retransmission attempts are less than or equal to the maximum allowed number of retransmissions—M_{ret}. To this end, it can be seen that, the ARQ algorithm serves similar reliability functions to the TCP one albeit at a different layer. Moreover, ARQ potentially increases, or at least causes fluctuations in the RTT of TCP, and this may interfere with the TCP timeout. Therefore, the assignment of ARQ parameters without having taken advantage of the information from the TCP state machine is suboptimal for the TCP flows.

In this regard, the main contributions of this chapter are as follows.

1. The research work presented in this chapter proposed a TCP-aware dynamic ARQ algorithm, which improves the end-to-end performance.

2. The proposed scheme utilises RTT of each TCP flow, to prioritise the (re)transmission of the packets so that unnecessary time out events can be avoided. As the RTT of the end-to-end paths are largely diverse, the achieved gain by using this technique can be significant.

3. The persistency of dynamic ARQ algorithm is also assigned depending on the RTO of TCP so that extra retransmissions over wireless link can be avoided, thus more efficient utilisation of the wireless channel can be achieved.

4. As the TCP-aware ARQ scheme introduces a new queuing discipline, queueing analysis is performed and the average waiting time in the transmission queues is computed.

5. Thorough investigations on the performance of the TCP-aware ARQ mechanism and its effects on the TCP are performed using the TCP implementation

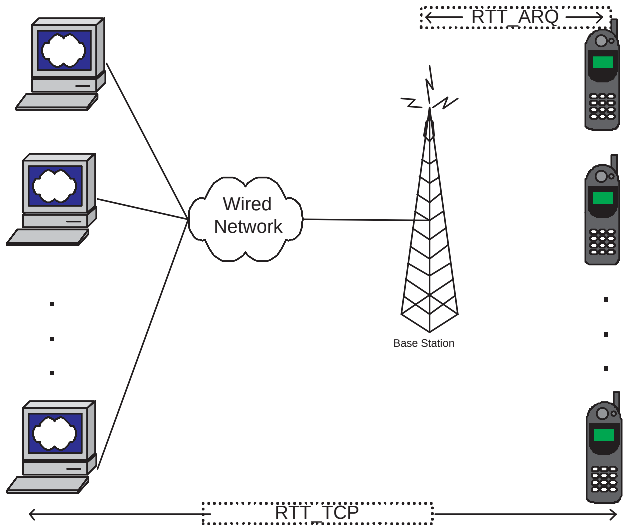


Figure 4.1: RTT over the wireless link is by far smaller than the end-to-end RTT.

within the OPNET modeler.

The remainder of this chapter is organised as follows. In the next Section, the related literature on the effect of ARQ mechanism on TCP performance is briefly reviewed. In Section 4.2, the details of the proposed TCP-aware dynamic ARQ protocol is presented. After elaboration on the queuing model of the dynamic ARQ protocol, Section 4.3 provides analysis to compute the queuing delay resulted from the proposed mechanism. Section 4.4 investigates the performance of the proposed scheme thoroughly. Finally, the remarking results and conclusion are detailed in Section 4.5.

4.1 Background Study

The number of (re)transmission attempts by an ARQ mechanism is called its persistency level. The ARQ mechanism, depending on its persistency, can be classified into three categories: Perfectly-Persistence, High-Persistence, and Low-Persistence [48]. Consequently, these three classes of Persistency results in a Fully-Reliable, Highly-Reliable, and Partially-Reliable link-layer. On the other hand, ARQ retransmissions increase delay in the packet delivery, thus augmenting the level of reliability can amplify this delay that is not desirable for many applications. Thereby, the tradeoff between reliability and delay exists in the design of an ARQ mechanism. This tradeoff is carefully discussed in RFC 3366 [48], and recommendations for the link-layer designers are given accordingly. Most of the designs for the link-layer, choose one of the last two classes of high/low-persistence ARQ, and find the number of (re)transmission attempts based on the constraints imposed by either wireless link or higher-layers.

In [49], the end-to-end loss rate is used to adapt the persistency level of the ARQ mechanism accordingly. Therefore, if the end-to-end connection experiences a high error rate on the wired part of the network, this algorithm provides lower reliability at the wireless link. On the other hand, the reliability is increased only if the error rate on the wired link also improves.

Another issue that needs to be discussed here, is the cross-layer interaction between TCP and link-layer such that timing information of TCP—e.g. RTT—can be available at the link-layer of wireless base station. These methods are detailed in Section 2.4.4, thus the repetition is avoided.

4.2 TCP-Aware Dynamic ARQ Mechanism

A cross-layer interaction between link-layer and TCP is defined to assign the persistency level of the ARQ mechanism on a per-flow basis. In addition, the priority of retransmissions is assigned based on the dynamics of TCP. An ultimate objective, is to achieve a TCP-aware dynamic ARQ algorithm, under the assumption that the information on the TCP timer is available at the wireless AP.

To this end, the TCP-aware ARQ mechanism assigns packets to the number of priority queues based on the number of reattempts of the corresponding packet. Afterwards, three functionalities are performed. First, in each of these queues packets are transmitted in the order of the remaining of their TCP flows' timer expiry.

Thus, packets have either waited longer in the transmission queue, or have shorter end-to-end paths, are retransmitted quicker.

Second, the expired packets by the TCP timer or the packets with the smaller remaining time than their actual RTT are dropped from the ARQ queues. ARQ mechanism stops retransmitting any undelivered packet, if the maximum allowed number of retransmissions is reached or the packet is dropped from the ARQ queues. Thus, in addition to the hard limit on the maximum number of retransmissions, which is a system parameter, a per packet limit also applies in this work. Finally, if the first high priority packet in queue $k+1$ is due to expire in less than two RTTs, this packet is moved to the front of queue k so as to be retransmitted first. To achieve the above detailed dynamic ARQ mechanism, a weighting parameter is associated with each packet. This mechanism is further discussed in the next section.

4.2.1 Algorithm Description

The non-ACK packets are categorised in priority queues, depending on their associated weighting parameters. The weighting parameter is assigned to each packet based on the remaining part of its retransmission time out counter denoted by T and the number of the so far retransmission attempts of the packet. Denoting by N, number of active TCP flows, and by P_{ij} the jth packet from TCP flow i, then w_{ij}, the weighting of P_{ij}, is described by,

$$w_{ij} = 10^4 \cdot m_{ij} + T_{ij}, \quad \forall i \in \{1, ..., N\}, \, \forall j \tag{4.1}$$

where m_{ij} is the number of reattempts of packet P_{ij}. The RTO of TCP is of the order of seconds (10^3 ms), thus the coefficient 10^4 normalises the weighting function value to the RTO, thus T in this equation is also in milliseconds.

In this respect, n queues are defined, namely $\{Q_1, ..., Q_n\}$. Based on this, packets are then assigned to each queue in accordance to their w_{ij}. Packet P_{ij} that locates in the position l of queue k is denoted by I_{kl}. Retransmissions start from Q_1 and

Algorithm 2 TCP-aware Dynamic ARQ Algorithm

a) For all i and j perform the following steps,

1. If $T_{ij} \leq RTT_i$, drop P_{ij}.
2. Else,
   Find $k \in \{1, ..., n\}$ such that w_{ij} satisfies $(k-1) \leq 10^{-4} \cdot w_{ij} \leq k$, and assign P_{ij} to Q_k.
3. In Q_k packets are sorted in the order of their weighting parameters.

b) For $k = 1$ to n

1. Transmit the first packet of Q_k.
2. For $K = k+1$ to n
   If the packet in I_{K1} has $T_{ij} \leq 2 \cdot RTT_i$, move P_{ij} to the front of Q_k.

are follow by retransmissions from the next queue, until all packets are transmitted. The key idea for implementing these priority queues is to avoid retransmitting any specific packet iteratively. Therefore, the residing queue of each packet is changed after each retransmission—this is applied by the effect of m_{ij} in the formulation of weighting parameters in Equation (4.1).

As discussed earlier, while transmitting from Q_k, if the timer expiry of the first packet in Q_{k+1}, $I_{k+1\,1}$, is occurring in less than two RTT_i, then this packet will be moved to Q_k and will be transmitted first. This later functionality, provides final retransmission chance over the wireless link for the packet that is close to being retransmitted by TCP. The full detail of the TCP-aware dynamic ARQ mechanism can be seen in Algorithm 2.

The information of RTT for each TCP flow can be known at the wireless AP using the techniques discussed in 2.4.4. Moreover, RTO of the corresponding TCP flow can be computed by the methods detailed in RFC 2988 [6] based on the average value of RTT and its deviation, D_{RTT}.

$$RTO = max(RTT + 4 * D_{RTT}, 1s). \tag{4.2}$$

4.3 Queuing Analysis of the TCP-Aware Dynamic ARQ Mechanism

In this section, queuing analysis is performed to estimate the queuing delay that results from the TCP-aware dynamic ARQ scheme. First, the queuing delay caused by the single queue ARQ mechanism is calculated, and afterwards, the delay due to the TCP-aware ARQ is derived. It has been mentioned in the previous section that the TCP-aware ARQ scheme introduces priority queues, thus the average waiting time of packets in this new queueing discipline is investigated.

The packets arriving at the link-layer, which are the newly arrived packets from the higher layer and also the non-acknowledged packets by the receiver that are queued to be retransmitted, are assumed to follow the Poisson distribution. Hence, the queuing delay can be analysed using either the $M/G/1$ or the $M/M/1$ queueing systems.

The link layer queues under consideration with a single server can be considered as an M/G/1 system with the average arrival rate λ. Given $\overline{x}$ and $\overline{x^2}$ the first two moments of service time consequently ($\overline{x}$ is also called the average service time), and the average waiting time in the queue can be described by the Pollacszek-Khinchin (P-K) formula as follows,

$$W = \frac{\lambda\overline{x^2}}{2(1-\rho)}, \tag{4.3}$$

where $\rho = \lambda\overline{x}$ is the queue utilisation. The derivation of the P-K formula assumes that customers are served in the order of their arrival. However, it can be shown that this formula is valid for any order of serving customers when this order is independent of the service time ([80], Chapter 3).

The queuing system of TCP-aware ARQ mechanism can be considered as an $M/G/1$ system such that arriving packets are divided into n different priority classes. A nonpreemptive priority rule is considered, thus the packet transmission is allowed to be completed without being interrupted by a newly arrived packet.

In this regard, the n priority queues are analysed to calculate the average queuing delay of the TCP-aware dynamic ARQ scheme. It is assumed that packets from the priority queue k arrive according to the Poisson distribution with rate λ_k, and their average service time is $\overline{x_k}$. Therefore, the following can be defined,

$$\lambda = \sum_{k=1}^{n} \lambda_k, \tag{4.4}$$

$$\overline{x} = \sum_{k=1}^{n} \frac{\lambda_k}{\lambda} \overline{x_k}, \tag{4.5}$$

$$\rho_k = \lambda_k \overline{x_k}, \tag{4.6}$$

$$\rho = \lambda \overline{x} = \sum_{k=1}^{n} \rho_k. \tag{4.7}$$

For packets in the queue with priority k, W_k denotes the average waiting time, and T_k denotes the total time in system. Waiting time of a packet in the queue k can be decomposed into three terms: delay that the packet encounters due to the packet found in service upon its arrival, delay caused by the packets that are already in the queues upon its arrival, and last, any delay due to the subsequent packets arrival.

The first term, denoted by W_0, can be defined as the average delay to any packet due to another packet in transmission, thus it can be expressed as,

$$W_0 = \sum_{k=1}^{n} \frac{\lambda_k \overline{{x_k}^2}}{2}, \tag{4.8}$$

where $\overline{{x_k}^2}$ is the second moment of service time for queue k.

Instead of addressing the second and third terms of W_k individually, the $M/G/1$ conservation law is illustrated to compute this delay. The $M/G/1$ conservation law implies that the weighted sum of the waiting times, is unchanged, no matter how elaborated the queueing discipline may be ([81], Chapter 3). Thus,

$$\sum_{k=1}^{n} \rho_k W_k = \begin{cases} \frac{\rho W_0}{1-\rho} & \rho < 1, \\ \infty & \rho \geq 1. \end{cases} \tag{4.9}$$

The conservation law puts a linear equality constraint on the set of average waiting times such that, attempt to reduce the waiting time in one queue results in the increase of some other W_ks—which is a rational argument. Given further the assumption that the service rate is equal on all the defined queues, i.e., $\overline{x_k} = \overline{x}$, the conservation law gives the average waiting time W' as follows,

$$W' = \frac{W_0}{1-\rho} = \sum_{k=1}^{n} \frac{\lambda_k \overline{x_k}^2}{2(1-\rho)}. \tag{4.10}$$

The average queueing delay of the simple ARQ mechanism, expressed by Equation (4.3), is equal to the average delay due to the priority queue discipline that is expressed by Equation (4.10).

In addition, the maximum delay arising from the priority queuing used in TCP-aware ARQ can also be calculated. This is the average delay that packets in queue Q_n experience, and can be expressed as ([80], Chapter 3),

$$W_n = \frac{W_0}{(1 - \sum_{k=1}^{n-1} \rho_k)(1 - \sum_{k=1}^{n} \rho_k)}. \tag{4.11}$$

Assuming sizes of the transmission packets follow an exponential distribution, and the wireless channel has Markovian model, the above discussed queues can be modelled as an $M/M/1$ queuing system. While this assumption holds, it can be shown that the average number of packets in the queuing system, are also equal for the both queueing discipline.

4.4 Performance Investigations

To observe how the performance of TCP is affected by the proposed TCP-aware dynamic ARQ mechanism, the proposed mechanism is implemented in the WLAN base station, thus simulations are performed using OPNET modeler. The commonly-used TCP Reno is assumed, and its Timestamp option is enabled that is incremented by one every 500 ms. This is well within the recommendation to increment by one at an interval of between 1 ms and 1 s ([60], Chapter 24). Moreover, link-level retransmissions are performed using the stop-and-wait ARQ algorithm.

In the considered scenarios, the WLAN nodes setup connections with wired servers via a single WLAN access point and wired routers. This is a similar configuration to the one depicted in Figure 4.1. The TCP-aware dynamic ARQ mechanism is implemented at the wireless AP, thus, Equation (4.1) associates weighting parameters with each packet prior to (re)transmission.

Simulation results are presented in terms of end-to-end throughput, comparing the proposed dynamic ARQ scheme with the normal 802.11a scheme, using a maximum of three retransmission attempts—four overall transmissions—in both cases. The overall results show that maximum value of the aggregated end-to-end throughput is improved approximately by $15-60\%$ in various conditions. The larger enhancements are when the mobile users are closer to the wireless AP, thus users experience better wireless channel.

4.4.1 Simulation Parameters

In the following simulation scenarios, the wireless channel is modelled with the path loss—PL α d^3 where d is the users' distance from the AP in meters—, Rayleigh fading—exponential random variable with $\beta = 1$—and LogNormal shadowing—standard deviation 4 dB. In each scenario, every mobile terminal connects to a unique server, via a wireless AP and the wired routers. The combination of traffic

Table 4.1: Simulation Parameters for the TCP-aware ARQ scheme

Simulation duration	600 s
FTP servers	16 MB file download size
HTTP servers	HTTP1.1 page interarrival time = exponential (mean 60 s) object size = exponential (mean 5 kB) number of objects per page = exponential (mean 7)
Email servers	Email size: 2 kB, Interarrival time = exponential (mean 120 s)
TCP MSS	1460 B
MAC buffer size	32 kB
MAC frame size	320 B
PHY characteristic	OFDM (IEEE 802.11a)
Operating frequency	5.4 GHz
Data Rate	6 Mbps

includes FTP, Web browsing and Email traffic, where in all scenarios, 40% of the clients are FTP users, 35% are HTTP users and the other 25% are Email users. More specific simulation parameters are given in Table 4.1.

4.4.2 Simulation Scenarios and Simulation Results

Simulation scenarios are designed to perform under various conditions of the end-to-end RTT distributions, traffic types, number and locations of mobile users in the cell. Therefore, in each of four scenarios, one parameter is changed in various runs of the simulation and the rest of the parameters are fixed. Locating the mobile users on a circle with radius d around the AP is similar to the depicted Figure 3.6 in Chapter 3.

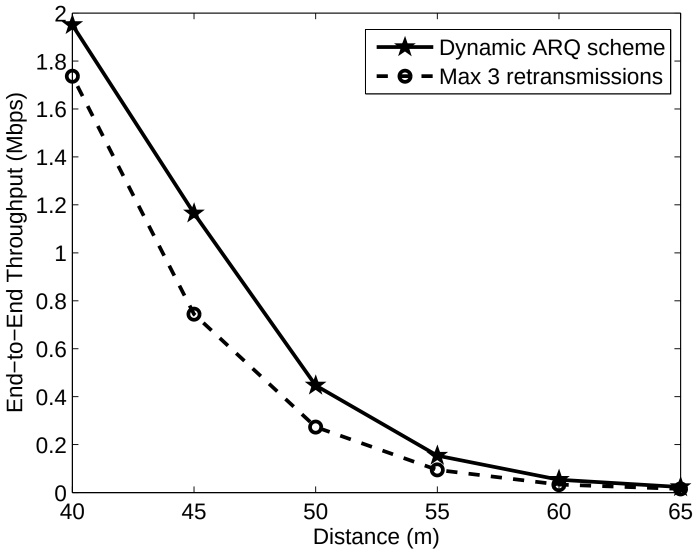


Figure 4.2: Scenario one: Aggregated end-to-end throughput versus wireless users' distance from the AP.

4.4.2.1 Simulation Scenario One: Changing the users' distance from the AP

In the first simulated scenario using 15 wireless clients, the RTT for each flow is a uniformly distributed random variable within range [5 ms, 100 ms]. Mobile users are located on a circle around the AP, where the radius of this circle is increased in each run of the simulation scenario in the range [40 m, 65 m]. Therefore, PL is the same for all the mobile users in each run. The achieved end-to-end throughput versus the mobile users' distance from the AP is plotted in Figure 4.2. The presented results here, show that the dynamic ARQ mechanism improves the end-to-end throughput in average by 15% for the range of mobile clients' location in the cell.

4.4.2.2 Simulation Scenario Two: Different distribution of the end-to-end RTTs

The second simulation scenario is performed over the 15 wireless clients that are mobile on a circle with radius 55 m around the wireless AP. Thus, the PL effect is constant during the whole simulation scenario and the received signal at the mobile

Table 4.2: Scenario two: End-to-End aggregated throughput for the Uniform, Normal and Exponentially distributed RTTs

RTT Distribution	Uniform	Normal	Exponential
Throughput (kbps): Dynamic	550	560	325
Throughput (kbps): 3. Ret	280	340	275

node is affected by the fading and shadowing parameters. The RTT values of the end-to-end paths are set according to different random distributions. These values are following the uniform distribution within [5 ms, 105 ms] in the first simulation run, and they are distributed normally with $\mu = 50$ ms, $\sigma = 20$ ms in the second run. Finally the exponential distribution with $\beta = 50$ ms is performed for the last run of this simulation scenario. The presented results in Table 4.2, show that the enhancement of throughput by the proposed dynamic ARQ scheme is largely unaffected by the specific RTT distribution.

4.4.2.3 Simulation Scenario Three: Various number of mobile users

In the third simulation scenario, given the same uniform distribution for the end-to-end paths' RTTs, wireless clients are on the circle with radius 55 m around the wireless AP. Moreover, the number of mobile users are increased in each simulation run from 5 to 20 mobile nodes. The results of this scenario that are presented in Figure 4.3, show the scalability of the dynamic ARQ scheme. In other words, a significant performance improvement of up to 60% is observed in the various number of mobile users competing over the wireless channel.

4.4.2.4 Simulation Scenario Four: Random distribution of users

The fourth simulated scenario is performed with the uniform distribution of 15 mobile users in the cell within distances of 40 m to 70 m from the wireless AP. The uniform distribution of users is performed using the method discussed in [77] and

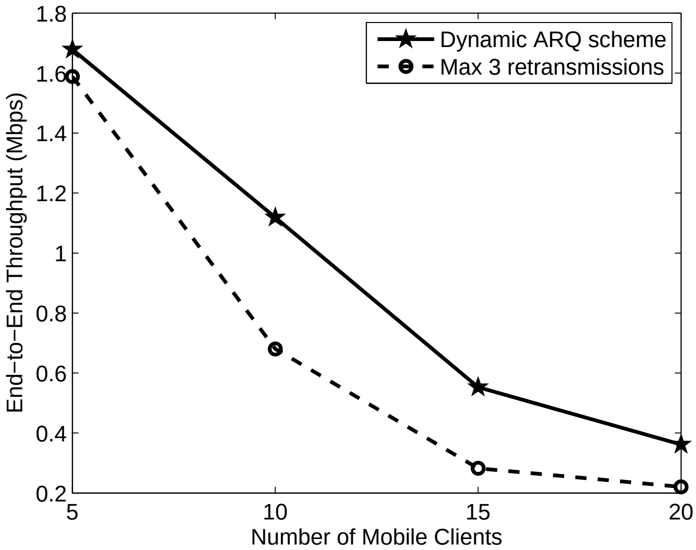


Figure 4.3: Scenario three: End-to-End aggregated throughput versus number of mobile users while users' distance from the AP is 55 m

also in Chapter 3. In this scenario, RTT values are following uniform distribution similar to the scenario three. The results of this scenario show 32% improvement in the end-to-end throughput.

4.4.2.5 Overall Results

With regard to the simulation results presented in Table 4.2 and Figures 4.2-4.3, the proposed TCP-aware dynamic ARQ algorithm improves TCP performance in the variety of scenarios. Packet loss experienced by TCP is decreased, and the end-to-end throughput is increased. Further observations reveal that, using the TCP-aware ARQ scheme affects the average number of retransmitted packets by TCP. These results that are presented in Table 4.3, show that the packet retransmission by TCP is decreased.

Moreover, the proposed algorithm is of low complexity from both the processing and buffering point of view. The highest complexity procedure is sorting the packets with regard to the weighting function—this complexity is $O(\log n)$. In addition, the TCP-aware ARQ scheme does not affect storage memory requirements of the link-layer.

Table 4.3: Average percentage of TCP retransmitted packets using the TCP-aware ARQ mechanism.

Simulated Scenario	1^{st}	2^{nd}	3^{rd}	4^{th}
Dynamic TCP-aware ARQ Scheme (ret. packets)	10%	9.7%	11.5%	10.4%
Three maximum Retransmissions (ret. packets)	11.6%	11.2%	13%	12%

4.5 Concluding Remarks

In this chapter, a TCP-aware ARQ mechanism has been proposed, which dynamically adapts the number of retransmissions and priority of transmission reattempts, based on RTT and RTO of the TCP flows, and the number of retransmissions which have thus-far been performed. A devised packet priority weighting function assists the retransmission process by receiving information as the cross-layer interaction between TCP and the link-layer. The TCP-aware dynamic ARQ mechanism changes the queueing discipline of the ARQ by introducing the priority queues. In this regard, the queuing delay resulted by the novel scheme is investigated. Analysis performed here show that, the average waiting time in the ARQ transmission queue, and the average number of packets in the system remains unchanged.

Finally, the proposed scheme is simulated using OPNET modeler, and its effect on the performance of real TCP implementations is examined. Extensive simulation scenarios are performed, and the results presented in Tables 4.2-4.3 and Figures 4.2-4.3 show a $15-60\%$ improvement in the end-to-end performance through this novel approach. It can also be seen that larger enhancements is achieved when mobile users are closer to the wireless AP, and as users move further from the AP, the improvement in the throughput is also decreased.

Chapter 5

TCP-Aware Resource Allocation Schemes

As the Internet grows both in terms of the number of users and diversity of applications, providing fair and efficient allocation of the available network resources becomes increasingly challenging. TCP is the default transport layer protocol used in the Internet to provide reliable end-to-end communications and is responsible for more than 90% of all Internet traffic. However, TCP exhibits a number of shortcomings when the underlying wireless medium deviates from the reliability of the wired medium for which TCP was originally designed to serve [39].

Despite the fact that TCP has been initially designed for elastic applications it is currently commonly used in various popular streaming applications. It is worthwhile noting that Real Media and Windows Media, the two dominant streaming media applications, are both based on TCP streaming. In that respect, in wireless networks where resources are scarce, TCP traffic for such applications should not be treated as best effort but some provision on the data rate has to be considered. In the proposed approach in this chapter, this provision is based on the theoretical average throughput that can be achieved by TCP, based on the specific path characteristics (i.e. RTT, PER).

The capacity of the end-to-end path is defined by the theoretical throughput of its corresponding TCP flow that depends on the end-to-end RTT and the PER as experienced by TCP. In this research work, the end-to-end achievable rate for different users is addressed by including the theoretical upper bound of the TCP throughput in the resource allocation problem in OFDMA. To this end, optimisation problems are formulated that attempt to maximise the overall achieved throughput at the wireless link with respect to appropriate resource allocation and at the same time, provide a balance towards TCP throughput. Considering that the theoretical TCP throughput is the highest steady-state throughput each end-to-end path can achieve, the rationale of the proposed approach is to distribute the resources more optimally, and fairer among the competing TCP flows.

In addition, the TCP connection is a bi-directional connection that requires ACK from the receiver for the transmitted data packets to achieve the reliability. This is particularly important for uplink resource allocation. Limited available bandwidth and congestion on the reverse path break down the principle of ACK clocking, and may cause an increase in the RTT [21]. Thus, TCP throughput on the forward path is degraded. Among several research works that explore these issues, some require explicit support from routers or middle nodes, whereas others are end-to-end schemes. In this chapter, the above discussed problem is addressed in the wireless network. The attempt is to bring the requirement of the TCP on the reverse path, into the actual radio resource allocation scheme. Therefore, the limited capacity on the uplink or congested uplink can be avoided. Unlike the existing end-to-end solutions [22], the proposed solution in this work is fully applied to the wireless resource allocation mechanism at the wireless AP and does not affect the end-hosts; in other words TCP remains unchanged.

The major contributions of this chapter are twofold. First, TCP-aware resource allocation scheme over downlink OFDMA is detailed. Two different formulations of the resource allocation problem are presented by using the theoretical TCP throughput as a mean of TCP-awareness in allocations, aiming to provide more balance allocations towards TCP throughput. It is revealed that as a result of such an allocation, fairness among end-to-end TCP flows is increased significantly. The second part investigates the problematic issue of scarce availability of bandwidth on the uplink, which has been widely studied in the asymmetric wired networks. Therefore, the joint uplink-downlink allocation problem is proposed not only to maximise the throughput on the downlink but also to guarantee the delivery of ACK packets on the uplink. To this end, the main contributions of this chapter are as follows,

1. The objective of the resource allocation problem in OFDMA is expanded to include the theoretical TCP throughput.
2. Resource allocation problem is defined over downlink OFDMA system aiming to enhance fairness among TCP flows while maximising the overall data rate. This scheme results in a more balanced throughput towards the TCP theoretical throughput.
3. To avoid the performance degradation of the end-to-end TCP flows due to the scarce resources in the uplink, a joint uplink-downlink resource allocation problem is proposed. This resource allocation scheme that is studied in the OFDMA-based wireless, guarantees to allocate the sufficient resources to the uplink of each individual TCP flow so that the allocated data rate on the downlink can be accomplished.
4. Through extensive simulation studies, it has been shown that the proposed TCP-aware resource allocation schemes enhance the performance of the end-to-end data transmission considerably. This performance is explored not only in terms of the achieved end-to-end throughput, but also in terms of fairness among TCP flows.

The remainder of this chapter is organised as follows. Section 5.1 provides an overview of the related literature in terms of OFDMA resource allocation scheme, existing TCP-aware resource allocation algorithms, and also the issue of asymmetric channel for the data delivery of TCP. In Section 5.2, the system model and baseline assumptions are detailed. After elaborating the constraints of the downlink resource allocation problem, Section 5.3 presents two alternatives of the optimisation problem for the downlink resource allocation scheme. Section 5.4 presents the joint uplink-downlink resource allocation problem that addresses the requirements of uplink in order to guarantee the delivery of the ACK packets for the allocated downlink. In Section 5.5, solving the proposed problem optimally with respect to the allocation of wireless resources is discussed. Section 5.6, presents heuristics to solve the proposed optimisation problems in real-time. After presenting the details of the simulation parameters, performance of the TCP-aware resource allocation problems are investigated in Section 5.7. Finally, Section 5.8 concludes this chapter.

5.1 Background Study

In this section the state of the art in the three related topics are given. Firstly, the variants of OFDMA resource allocation schemes are addressed. Secondly, the existing TCP-aware resource allocation algorithms in the literature are discussed. Finally, an overview of the investigations on the performance of TCP over asymmetric links is given.

5.1.1 OFDMA Resource Allocation Schemes

Next generation wireless technologies, such as LTE [82] (in the downlink) and IEEE 802.16e [35], specify OFDMA as their access method. OFDMA divides the available bandwidth into multiple orthogonal subcarriers, allowing users to transmit simultaneously through allocating different subsets of the available subcarriers to different users. It is well known that dynamic allocation of subcarriers can significantly improve the overall performance of OFDMA systems. Thus, the implied problem of the joint subcarrier assignment and resource (bits and power) allocation for the OFDMA has been a prominent area of research over the past few years.

Much of the past research works in the literature have concentrated on objectives such as maximising the overall data rate subject to power or BER constraints [83]. Rate maximisation problem is more relevant for data centric networks. The dual problem can address minimising overall power consumption in the interest of energy efficiency with the minimum rate constraint [84]. This problem is of more interest for the applications requiring fixed data rate. However, none of the above approaches can satisfy fairness among users.

Alternative formulations do, however, consider fairness, either by prioritisation using the weighted sum rate method [85], or by introducing proportional rate constraints [86]. A further approach is presented in [87], in which fairness is considered by maximising the lowest achieved data rate among the user set. The research presented in [88] addresses proportional fairness in OFDMA resource allocations, based on the Nash bargaining solution. The resource allocation problem that maximises sum rate but also satisfy per flow QoS requirements is presented in [89]. Although these research works among many others investigate the issues of fairness and QoS with respect to the allocated data rate over the wireless link, aspects related to the end-to-end data transmission perspective have not been sufficiently addressed.

5.1.2 TCP-Aware Resource Allocation Schemes

A thorough overview of cross-layer design for resource allocation algorithms in the third generation wireless networks is given by [56], where TCP over CDMA is also addressed. TCP-aware resource allocation algorithms over a CDMA network are studied in [90], the objective being to maximise throughput. The novel proposed algorithm in that paper uses information on the TCP state to allocate the data rate more appropriately at the wireless link. A joint congestion control and power allocation in a CDMA based wireless network is proposed in [58], in which a generalised network utility maximisation framework is also presented. In the context of IEEE 802.16, reference [59] proposes a TCP-aware allocation algorithm which estimates the bandwidth demand based on the long-term data rate, and allocates resources accordingly. Unlike available solutions in the literature, in this work, the closed form expression of TCP throughput is used as a means of TCP-awareness in the allocations, and in contrast with existing TCP-aware resource allocation techniques, the focus is on OFDMA-based systems.

5.1.3 TCP over Asymmetric Links

The effect of link asymmetry on the performance of TCP is widely studied in wired networks. Limited available bandwidth and congestion on the reverse path break down the principle of ACK clocking, and may cause an increase in the RTT [21]. Thus, TCP throughput on the forward path is degraded. In this regard, several research works explore these issues and a range of solutions are proposed. Some of these proposals require explicit support from routers or middle nodes, whereas others are end-to-end schemes. For example, ACK congestion control [22] is one of the solutions that attempts to reduce the sending rate for ACK traffic, with the assumption that the reduction in ACK rate may help to cut the congestion itself. In this research work, the above discussed issue is explored in wireless networks. Unlike the existing end-to-end solutions [22], the proposed solution in this work is fully applied to the wireless resource allocation mechanism at the wireless AP and does not affect the end-hosts; in other words TCP remains unchanged.

5.2 System Model

The n active TCP flows are assumed, all of which operate in congestion avoidance phase. A single cell OFDMA network is assumed with m available subcarriers. Let for flow i the rate on subcarrier j to be r_{ij}. Each user is associated with a single TCP flow, therefore, the achievable rate for user i can be written as follows,

$$R_i = \sum_{j=1}^{m} a_{ij} r_{ij}, \tag{5.1}$$

where,

$$a_{ij} = \begin{cases} 1 & \text{if subcarrier } j \text{ is assigned to user } i, \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

The channel gain of user i at subcarrier j is denoted by G_{ij}. With the thermal noise power, σ^2, the ith user's received Signal to Noise Ratio (SNR) on subcarrier j is thus denoted as,

$$\gamma_{ij} = \frac{p_{ij} G_{ij}}{\sigma^2}. \tag{5.3}$$

where p_{ij} is the allocated power to flow i on subcarrier j.

Adaptive modulation provides the desired rate in the allocated subcarrier for each individual user. Given $c_1 \approx 0.2$, $c_2 \approx 1.5$, BER is expressed based on the adaptive M-QAM. [91].

$$BER_{ij} \approx c_1 \, e^{-c_2 \frac{\gamma_{ij}}{2^{r_{ij}}-1}}. \tag{5.4}$$

Similar to [88], the same, fixed, BER for all users in all subcarriers is assumed i.e. $BER_{ij} = b \;\; \forall i, j$. Given $c_3 = -ln(b/c_1)/c_2$, and solving for r_{ij}, the achievable rate for user i on the jth subcarrier can be described as follows,

$$r_{ij} = w_j \log_2 \left(1 + \frac{p_{ij} G_{ij}}{\sigma^2 c_3} \right) \;\; bits/s. \tag{5.5}$$

In Equation (5.5), w_j is the bandwidth of subcarrier j which is assumed to be equal for all subcarriers and will be denoted hereafter by w. The wireless channel suffers from slow-fading effect such that the channel is constant within each OFDM frame. The slowly time varying assumption is crucial since it is also assumed that perfect estimation of the subchannels is available for each user. Moreover, mobile users and the base station are synchronised, thus there is negligible inter-carrier interference.

5.3 Downlink Resource Allocation Problem

In data centric networks, a common objective of various resource allocation problems is to maximise the overall throughput. Moreover, user requirements such as fairness and QoS provisioning can also be implicitly or explicitly considered in such a setting. On the other hand, utilising side information such as the minimum data rate requirements, or the maximum achievable data rate for each user can provide significant benefits on the design of more efficient resource allocation strategies. In this respect, the aim of the proposed downlink resource allocation problem here, is to determine the users' transmission functions $[A]_{ij} = a_{ij}$ and power matrix $[P]_{ij} = p_{ij}$ in order to maximise the overall rate with regard to the power constraints while TCP fairness among the active end-to-end flows [92] is also satisfied. Two formulations of the downlink TCP-aware resource allocation problem are detailed as follows.

5.3.1 Proportional TCP Throughput Constrained (P1)

The formulated optimisation problem (P1) aims to maximise the downlink sum rate, while TCP fairness is assured by imposing a set of nonlinear constraints into the problem. Thus, the proportional downlink rate among users are constrained with respect to the TCP theoretical throughput. The proposed resource allocation scheme can be summarised in the following problem,

$$\text{(P1)}: \text{ Maximise } \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} w \log_2\left(1 + \frac{p_{ij}G_{ij}}{\sigma^2 c_3}\right),$$

$$\text{subject to: } \sum_{i=1}^{n} a_{ij} \leq 1, \qquad \forall j \in \{1, ..., m\} \tag{5.6}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} p_{ij} \leq P_T, \tag{5.7}$$

$$\frac{R_i}{B_i} = \frac{R_1}{B_1}, \qquad \forall i \in \{2, ..., n\} \tag{5.8}$$

$$p_{ij} \geq 0, \qquad \forall i \in \{1, ..., n\}, j \in \{1, ..., m\} \tag{5.9}$$

$$a_{ij} \in \{0,1\}. \qquad \forall i \in \{1,...,n\}, j \in \{1,...,m\} \tag{5.10}$$

where P_T is the total available power at the base station. Constraint (5.6) imposes that every subcarrier is assigned to one user, and constraint (5.7) restricts the total available power at the base station. Moreover, constraint (5.8) provide fairness among TCP flows with maintaining proportional rate with respect to the TCP theoretical throughput for each user. The expressions for TCP throughput are the same as the models detailed in Section 2.1.4. Finally, constraints (5.9) and (5.10) ensure the correct assigned values for the power and subcarrier allocation matrices.

5.3.2 Rate Difference from TCP Throughput Constrained (P2)

The second resource allocation problem investigates the difference between the allocated wireless link data rate and the theoretical achievable TCP throughput. Given D_i, the difference between allocated rate to the ith user and the achievable rate by TCP flow i, the aim is to maximise the overall rate while minimising the D_is.

$$D_i = \mid \alpha \cdot B_i - R_i \mid . \tag{5.11}$$

In Equation (5.11), α represents the proportion between the throughput at the TCP layer and at the physical layer, which is the result of TCP/IP headers. Therefore, the resource allocation problem can be formulated as follows,

$$\text{(P2)}: \text{ Maximise} \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} w_j \log_2\left(1 + \frac{p_{ij}G_{ij}}{\sigma^2 c_3}\right) - \mu \sum_{i=1}^{n} D_i,$$

$$\text{subject to: } \sum_{i=1}^{n} a_{ij} \leq 1, \qquad \forall j \in \{1,...,m\} \tag{5.12}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} p_{ij} \leq P_T, \tag{5.13}$$

$$p_{ij} \geq 0, \qquad \forall i \in \{1, ..., n\}, j \in \{1, ..., m\} \tag{5.14}$$

$$a_{ij} \in \{0, 1\}. \qquad \forall i \in \{1, ..., n\}, j \in \{1, ..., m\} \tag{5.15}$$

Constraints (5.12)- (5.15) are similar to (5.6)- (5.7) and (5.9)- (5.15).

The addressed problem is a multi objective optimisation problem. There are various approaches towards solving such a problem. The well-studied approach to combine the multiple objectives into a single objective is used here whose solution is Pareto optimal. Therefore, the optimal solution is not unique, and it depends on the value of μ. Selecting the value of μ balances the two objectives. In problem (P2), increasing μ can move the allocation balance towards TCP throughput, while decreasing μ move the balance towards data rate maximisation. In Section 5.7, the effect of changing μ on the performance of the proposed scheme is examined.

5.4 Joint Uplink-Downlink Resource Allocation Problem

In data centric networks, it is assumed that downlink carries the mass of traffic. Thus, much of the past works in the literature have concentrated on the allocation of resources in the downlink. On the other hand, depending on the duplexing method, the available resources are divided either in time or in frequency between downlink and uplink channels. The border, d, depicted in Figure 5.1, can be defined as one of the system parameters; e.g. in the LTE assumptions [82] the uplink capacity is equal to half the downlink capacity. Moreover, d can be defined dynamically based on various system constraints to guarantee the requested QoS. In this research work, d is defined dynamically, aiming to satisfy requirements of the end-to-end TCP connections.

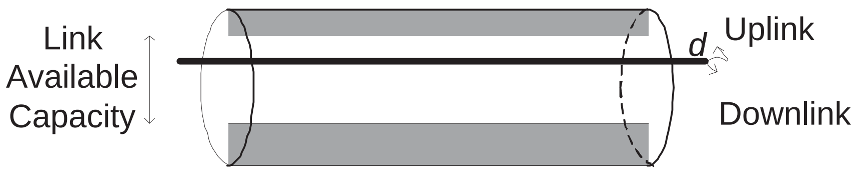


Figure 5.1: Downlink and uplink capacity of the wireless link

5.4.1 Uplink Requirements

With the increasing number of applications uploading data e.g., Emails with large attachments, the uplink resources can become more scarce. TCP connection as a bi-directional connection, requires ACK from the receiver for the transmitted data to achieve the reliability. Therefore, the effect of link asymmetry on the performance of TCP, which has been widely studied in the wired networks [93], can be also crucial in wireless networks.

To depict a more clear picture, an example is detailed here, in which it is assumed that a user downloads data over a link with 20 Mbps capacity, while the uplink capacity is limited to 100 kbps. If the lengths of data packets are 1500 B and the lengths of ACK packets are 40 B, TCP can only send ACKs for every five packets, otherwise the uplink path will be saturated. Therefore, the principle of ACK clocking can break down, and the sender clocks out new data at a slower rate. In other words the sender cwnd grows slower and TCP flow utilises the allocated downlink bandwidth inefficiently.

In the above example, if TCP acknowledges every single packet, it can achieve not more than 4 Mbps on the downlink. This phenomena can be seen in Figure 5.2, where the downlink throughput is plotted versus the link BER. The presented results in Figure 5.2 are simulated using OPNET modeler in which the end-to-end RTT is 60 ms, the downlink data rate is 20 Mbps, and the uplink data rate is decreased from 1 Mbps to 100 kbps, and then to 10 kbps. The packet size and ACK size are assumed to be the same as the previous example.

Thereby, the joint downlink-uplink problem can be defined, not only to provide the

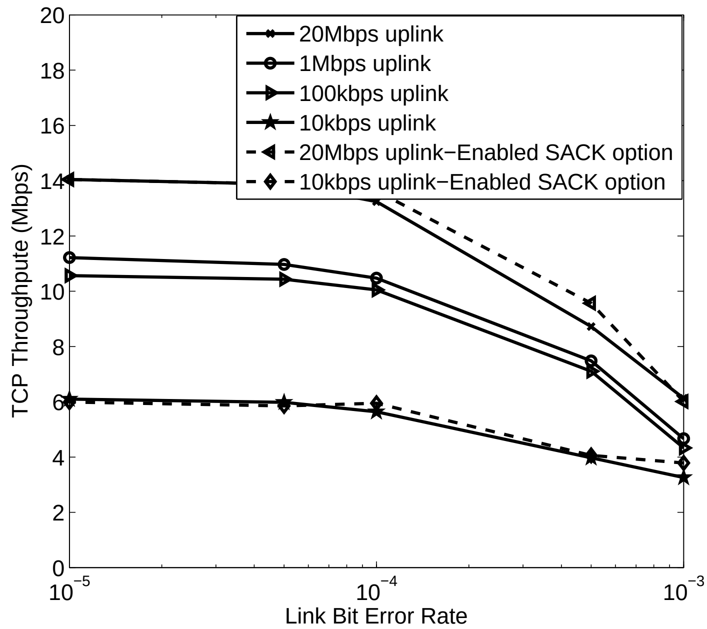


Figure 5.2: Effect of link asymmetry on the TCP throughput: Downlink throughput versus link BER, downlink capacity is 20 Mbps

optimal allocation on the downlink, but also to guarantee the delivery of downlink packets with the appropriate data rate on the uplink.

5.4.2 Proportional TCP Throughput Constraint (P3)

To address the requirement of the TCP connection on the uplink, the optimisation problem that is formulated here is a similar problem to that in (P1), but also the minimum rate requirement applies to the uplink resource allocation. Thus, the joint optimisation problem (P3) aims to maximise the aggregated rate on the downlink while providing a sufficient proportion of the downlink data rate for the uplink.

Assuming TCP acknowledges every single packet, the minimum required data rate on the uplink would be a proportion of the downlink depending on the size of the ACK packet, which is the TCP header size, thus $R_{u_i} \geq \rho R_{d_i}$, $0 < \rho < 1$. This proportion can be increased for example with using the SACK option.

Hence, the resource allocation scheme would be summarised in the following problem,

$$(\text{P3}) : \text{Maximise} \sum_{i=1}^{n}\sum_{j=1}^{m} c_j a_{ij} w \log_2\left(1+\frac{p_{ij}G_{ij}}{\sigma^2 c_3}\right),$$

subject to:

$$\sum_{i=1}^{n} a_{ij} \leq 1, \qquad \forall j \in \{1,...,m\} \tag{5.16}$$

$$\sum_{j=1}^{m} c_j \leq m_d, \qquad m_d \in \{1,2,...,m\}. \tag{5.17}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} c_j a_{ij} p_{ij} \leq P_T, \tag{5.18}$$

$$\sum_{j=1}^{m} (1-c_j)\, a_{ij} p_{ij} \leq P_t, \qquad \forall i \in \{1,...,n\} \tag{5.19}$$

$$\frac{R_{d_i}}{B_i} = \frac{R_{d_1}}{B_1}, \qquad \forall i \in \{2,...,n\} \tag{5.20}$$

$$R_{u_i} \geq \rho R d_i, \qquad \forall i \in \{1,...,n\} \tag{5.21}$$

$$p_{ij} \geq 0, \qquad \forall i \in \{1,...,n\}, j \in \{1,...,m\} \tag{5.22}$$

$$a_{ij} \in \{0,1\}, \qquad \forall i \in \{1,...,n\}, j \in \{1,...,m\} \tag{5.23}$$

$$c_j \in \{0,1\} \qquad \forall j \in \{1,...,m\} \tag{5.24}$$

Let c_j represent the allocation of subcarrier j to either downlink (c_j=1) or uplink (c_j=0). Thereby, constraint (5.17) bounds the number of downlink subcarriers to m_d, and the optimal value can be found solving problem (P3). Constraints (5.18) and (5.19) restrict the total available power at the base station, P_T, and at each mobile user, P_t. Moreover, constraint (5.21) provides the required data rate for uplink, in order to guarantee delivery of the downlink allocated resources.

5.4.3 Rate Difference from TCP Throughput Constraint (P4)

A similar approach to the above in defining problem (P3) is taken here, in order to convert the TCP-aware downlink resource allocation (P2) to the joint uplink-downlink resource allocation scheme. Thus, problem (P4) is defined to minimise the gap between the allocated data rate and the theoretical TCP throughput for each TCP flow on the downlink. Moreover, it satisfies the minimum uplink rate constraint of each TCP flow, which is expressed as follows,

$$\text{(P4)} : \text{ Maximise } \sum_{i=1}^{n}\sum_{j=1}^{m} c_j a_{ij} w \log_2\left(1+\frac{p_{ij}G_{ij}}{\sigma^2 c_3}\right) - \mu\sum_{i=1}^{n} D_i \,.$$

$$\text{subject to: } \sum_{i=1}^{n} a_{ij} \le 1, \qquad \forall j \in \{1,...,m\} \tag{5.25}$$

$$\sum_{j=1}^{m} c_j \le m_d, \qquad m_d \in \{1,2,...,m\}, \tag{5.26}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} c_j a_{ij} p_{ij} \le P_T, \tag{5.27}$$

$$\sum_{j=1}^{m} (1-c_j)\, a_{ij} p_{ij} \le P_t, \qquad \forall i \in \{1,...,n\} \tag{5.28}$$

$$R_{u_i} \ge \rho R d_i, \qquad \forall i \in \{1,...,n\} \tag{5.29}$$

$$p_{ij} \ge 0, \qquad \forall i \in \{1,...,n\}, j \in \{1,...,m\} \tag{5.30}$$

$$a_{ij} \in \{0,1\}, \qquad \forall i \in \{1,...,n\}, j \in \{1,...,m\} \tag{5.31}$$

$$c_j \in \{0,1\}. \qquad \forall j \in \{1,...,m\} \tag{5.32}$$

Constraints (5.25)-(5.32) are the same as (5.16)-(5.19) and (5.21)-(5.24).

5.5 Optimal Subcarrier Allocation and Power Distribution

Clearly, subcarrier and power should be assigned jointly to achieve the optimal solution. This joint allocation represents a mixed integer non-linear mathematical programming problem which pose a high computational complexity. As an example, the optimal solution for optimisation problem (P1) is detailed here, hence similar approach can be used to solve the other three optimisation problems.

To solve the optimisation problem (P1) optimally, it can be reformulated to a continuous problem by defining discrete variable a_{ij} as a continuous variable over the region $[0, 1]$. Moreover, the objective function needs to be rewritten as,

$$\sum_{i=1}^{n} R_i - M \cdot \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}(1 - a_{ij}), \tag{5.33}$$

where M is a relatively large value to ensure the integer assignment for a_{ij}. For problems (P3) and (P4) similar conversion is required to be applied over the discrete variable c_{ij}.

The continuous reformulation of problem (P1) can be solved using well-known methods and also optimisation toolboxes. On the other hand, by increasing the number of mobile users and the number of subcarriers, the number of constraints of the optimisation problem (P1) are also increased. Thus, solving the problem is computationally complex, and it can be prohibitive for the base station to solve this problem in real-time.

For real-time implementation and to allow larger instances of the problem to be solved, a greedy allocation is presented that provides suboptimal but feasible solutions. Thus, the optimisation problem is decoupled first to allocate set of subcarriers to each user and afterwards, to allocate power to the certain set of subcarriers. This solution is suboptimal, and is further discussed in the next section.

5.6 Suboptimal Subcarrier Allocation and Power Distribution

To decouple the proposed optimisation problems, the approach similar to [86] is taken. In the subcarrier allocation it is assumed that power is equally distributed in all the subcarriers, therefore the solution is suboptimal. Afterwards, for a certain subcarrier allocation, the optimisation problem can be reformulated over the contin-

Algorithm 3 Subcarrier Allocation to the Optimisation Problem (P1)

a) Initialisation

1. Set R_i=0 and $\Omega_i = \phi$ for i=1 to n and C={1,2,...m}.
2. Sort the users' index in the descending order of B_i.

b) for i=1 to n

1. Find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C$.
2. Let $\Omega_i = \Omega_i \cup \{k\}$ and $C = C - \{k\}$.
3. Update R_i

c) while $C \neq \phi$

1. Find user l satisfying $\frac{R_l}{B_l} < \frac{R_i}{B_i}$ for all $i \in \{1, ..., n\}$.
2. For user l, find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C$.
3. Let $\Omega_l = \Omega_l \cup \{k\}$ and $C = C - \{k\}$.

uous variable p_{ij}. Thus, using the water filling approach, power will be distributed optimally with respect to maximising the cell data rate.

5.6.1 Downlink Resource Allocation Scheme (P1)

The subcarrier allocation for the optimisation problem (P1) is performed in the few iterations as follows. The principle of the algorithm is for each user to allocate the subcarrier with the highest channel gain available. At the first iteration, each user selects the best available subcarrier, starting from the user with the highest value of B_i and continue in the order of their B_i values. Afterwards, at each iteration, the user with the lowest proportion of $\frac{R_i}{B_i}$ has the option to choose the subcarrier. Finally, Ω_i is the set of assigned subcarriers to user i. These iterative steps are detailed in Algorithm 3.

In the next step, to a certain subcarrier allocation, problem (P1) will be simplified into a maximisation problem over continuous variable p_{ij}.

$$(\mathrm{P1}') : \text{Maximise} \sum_{i=1}^{n} \sum_{j \in \Omega_i} w_j \log_2 \left(1 + \frac{p_{ij} G_{ij}}{\sigma^2 c_3}\right),$$

$$\text{subject to:} \sum_{i=1}^{n} \sum_{j \in \Omega_i} p_{ij} \leq P_T, \tag{5.34}$$

$$\frac{R_i}{B_i} = \frac{R_1}{B_1}, \qquad \forall i \in \{2, ..., n\} \tag{5.35}$$

$$p_{ij} \geq 0. \qquad \forall i \in \{1, ..., n\}, j \in \{1, ..., m\} \tag{5.36}$$

where Ω_i is the set of assigned subcarrier to user i such that, Ω_{i_1} and Ω_{i_2} are mutually exclusive if $i_1 \neq i_2$.

The problem of power distribution among subcarriers, and the performance comparison between equal and optimal power distribution is well-studied in [94], in which it has been shown that equal power distribution can not generally result in near optimal solution. Therefore, to find the optimal solution for power distribution, similar approach to [88] and [86] is used. Thus, problem (P1$'$) can be solved writing the lagrangian dual function. Differentiating the lagrangian dual function with respect to p_{ij} and set the derivative to zero, power can be distributed with the same method presented in [86].

Problem (P1$'$) can be solved writing the lagrangian dual function.

$$\begin{aligned} L_1 = & \sum_{i=1}^{n} \sum_{j \in \Omega_i} w \log_2 \left(1 + \tfrac{p_{ij} G_{ij}}{\sigma^2 c_3}\right) + \left(\sum_{i=1}^{n} \sum_{j \in \Omega_i}^{m} -\nu_{ij} p_{ij}\right) \\ & + \lambda \left(\sum_{i=1}^{n} \sum_{j \in \Omega_i} p_{ij} - P_T\right) \\ & + \sum_{i=2}^{n} \eta_i \left(\sum_{j \in \Omega_1} w \log_2 \left(1 + \tfrac{p_{1j} G_{1j}}{\sigma^2 c_3}\right) - \sum_{j \in \Omega_i} \tfrac{B_1}{B_i} w \log_2 \left(1 + \tfrac{p_{ij} G_{ij}}{\sigma^2 c_3}\right)\right). \end{aligned} \tag{5.37}$$

In Equation (5.37), ν_{ij}, λ, γ_i, η_i, and ξ_i are lagrangian multipliers. Differentiating the lagrangian dual function with respect to p_{1j} and p_{ij} and set the derivative to

Algorithm 4 Subcarrier Allocation to the Optimisation Problem (P2)

a) Initialisation

1. Set R_i=0 and $\Omega_i = \phi$ for i=1 to n and C={1,2,...m}.
2. Sort the users' index in the descending order of B_i.

b) for i=1 to n

1. Find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C$.
2. Let $\Omega_i = \Omega_i \cup \{k\}$ and $C = C - \{k\}$.
3. Update R_i

c) while $C \neq \phi$

1. Find user l satisfying $R_l - \mu D_l < R_i - \mu D_i$ for all $i \in \{1, ..., n\}$.
2. For user l, find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C$.
3. Let $\Omega_l = \Omega_l \cup \{k\}$ and $C = C - \{k\}$.

zero, power can be distributed similar to [86] based on the water-filling algorithm.

5.6.2 Downlink Resource Allocation Scheme (P2)

The suboptimal solution to (P2) is detailed here, where at each iteration, the user with the lowest value of $R_i - \mu D_i$ selects a subcarrier. The main principle of this algorithm is similar to the presented solution for (P1) such that, at each round the available subcarrier with the highest channel gain is allocated to the user. Finally, through iterative steps detailed in Algorithm 4, Ω_i is the set of assigned subcarriers to user i.

Based on a pre-defined subcarrier allocation, problem (P2) is simplified into a maximisation problem over the continuous set of variables p_{ij}.

$$(\text{P2}') : \text{ Maximise } \sum_{i=1}^{n} \sum_{j \in \Omega_i} w_j \log_2 \left(1 + \frac{p_{ij} G_{ij}}{\sigma^2 c_3}\right) - \mu \sum_{i=1}^{n} D_i,$$

subject to:

$$\text{subject to: } \sum_{i=1}^{n} \sum_{j \in \Omega_i} p_{ij} \leq P_T, \tag{5.38}$$

$$p_{ij} \geq 0. \qquad \forall i \in \{1, ..., n\}, j \in \{1, ..., m\} \tag{5.39}$$

Similar to problem (P1′), optimisation problem (P2′) can be solved using the water-filling approach.

5.6.3 Joint Uplink-Downlink Resource Allocation Scheme (P3)

In the joint resource allocation problem (P3), a similar approach to problem (P1) is used to allocate the downlink subcarriers, using an initial value for m_d. The initial value of m_d is selected such that number of uplink subcarriers are ρ times the number of downlink subcarriers, $m_d = m \cdot 1/1 + \rho$. Afterwards, in few iterations, the largest value of m_d which satisfies constraint (5.21) will be found. This value clearly is the optimal value to maximise the objective function. Moreover, the uplink subcarriers are allocated using the same principle as downlink, such that the available subcarrier with the highest channel gain is first allocated to the user. The above procedure is detailed in Algorithm 5, in which Ω_i is the set of allocated subcarriers to the user i in the downlink and Ψ_i is the set of allocated subcarriers to this user in the uplink. In the Algorithms 5, Ω_{i_1} and Ω_{i_2} are mutually exclusive, if $i_1 \neq i_2$. The flow chart of Algorithm 5 is also depicted in Figure 5.3.

The problem of power allocation to a certain subcarrier allocation, is as follows,

$$(\text{P3}') : \text{ Maximise } \sum_{i=1}^{n} \sum_{j \in \Omega_i} w \log_2 \left(1 + \frac{p_{ij} G_{ij}}{\sigma^2 c_3}\right),$$

subject to:

$$\sum_{i=1}^{n}\sum_{j\in\Omega_i} p_{ij} \leq P_T, \tag{5.40}$$

$$\sum_{j\in\Psi_i} p_{ij} \leq P_t, \qquad \forall i \in \{1,...,n\} \tag{5.41}$$

$$\frac{R_{d_i}}{B_i} = \frac{R_{d_1}}{B_1}, \qquad \forall i \in \{2,...,n\} \tag{5.42}$$

$$R_{u_i} \geq \rho R_{d_i}, \qquad \forall i \in \{1,...,n\} \tag{5.43}$$

$$p_{ij} \geq 0. \qquad \forall i \in \{1,...,n\}, j \in \{1,...,m\} \tag{5.44}$$

5.6.4 Joint Uplink-Downlink Resource Allocation Scheme (P4)

Subcarrier allocation to (P4) is similar to subcarrier allocation to (P2), with adding the uplink allocation. Number of downlink subcarriers, m_d, is defined in the set of iterations as discussed for problem (P3). Algorithm 6, presents the detail of subcarrier allocation to optimisation problem (P4).

Problem (P4) can also be rewritten as (P4′) over the continuous variable p_{ij}.

$$(\text{P4}') : \text{ Maximise } \sum_{i=1}^{n}\sum_{j\in\Omega_i} w\log_2\left(1+\frac{p_{ij}G_{ij}}{\sigma^2 c_3}\right) - \mu\sum_{i=1}^{n} D_i,$$

$$\sum_{i=1}^{n}\sum_{j\in\Omega_i} p_{ij} \leq P_T, \tag{5.45}$$

$$\sum_{j\in\Psi_i} p_{ij} \leq P_t, \qquad \forall i \in \{1,...,n\} \tag{5.46}$$

$$R_{u_i} \geq \rho R_{d_i}, \qquad \forall i \in \{1,...,n\} \tag{5.47}$$

$$p_{ij} \geq 0. \qquad \forall i \in \{1,...,n\}, j \in \{1,...,m\} \tag{5.48}$$

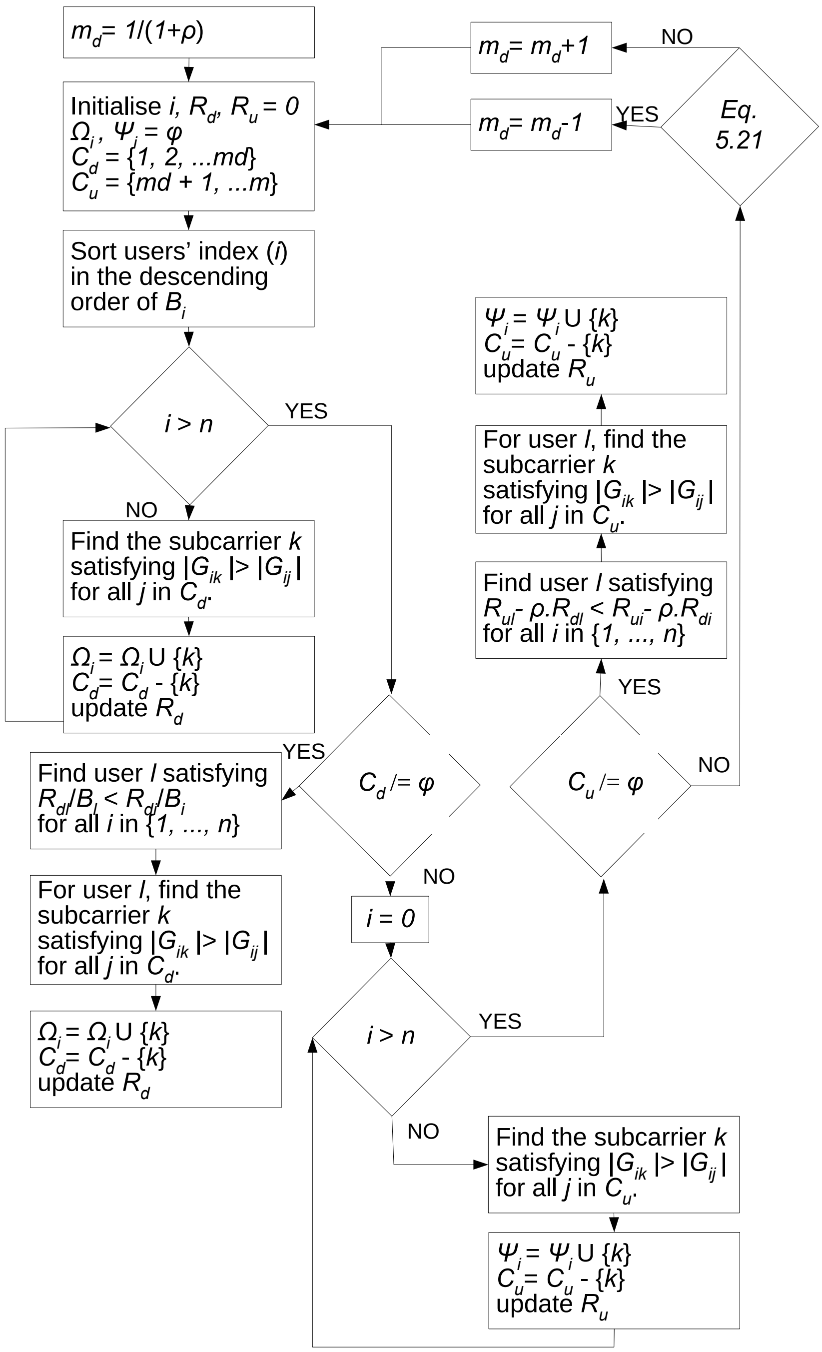


Figure 5.3: The flow chart that corresponds to Algorithm 5.

5.7 Performance Investigations

The performances of both downlink and joint uplink-downlink resource allocation schemes are explored in this section.First, the performance of downlink resource allocation problem is investigated, thus improvement in the fairness among TCP flows is observed. Afterwards, the proposed joint uplink-downlink problem is examined, in which the enhancement in the aggregated throughput can also be observed.

5.7.1 Fairness Analysis

Jain's fairness index [67], denoted by FI, is used to measure fairness of the downlink allocation scheme among TCP flows. This index is well-used as a quantitative measure of fairness in both wired and wireless networks. The index FI is 1 when there is a complete fair allocation.

$$FI = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot \sum_{i=1}^{n} {x_i}^2}. \tag{5.49}$$

Assuming x_i is the data rate of user i, proportional to the optimal rate that can be achieved on the corresponding end-to-end path, then FI as described in (5.49) can be the measure of fairness among end-to-end flows. The optimal throughput for each TCP flow, is the theoretical TCP throughput—detailed in Section 2.1.4.

5.7.2 Simulation Parameters

In order to investigate performance of the proposed scheme a number of different scenarios are considered. An OFDMA system with 52 subcarriers is simulated—this is equal to the number of OFDM subcarriers in IEEE 802.11a. The rest of the simulation parameters are similar to those used in [82], which are also summarised

Table 5.1: Simulation Parameters for the TCP-aware Resource Allocation Scheme.

Bandwidth	5 MHz
Target BER	10^{-4}
Channel model	ITU Pedestrian B
Shadowing standard deviation	8 dB
Average SNR	15 dB
Total power at the base station	43 dBm
Total power at the mobile user	23 dBm

in Table 5.1. The available bandwidth is 5 MHz, maximum available power at the base station is 43 dBm, and at each mobile user is 23 dBm. The thermal noise power, σ^2, is -107 dBm (Johnson-Nyquist noise over 5 MHz bandwidth), and the target BER is 10^{-4}. The wireless channel is modelled with ITU pedestrian model, and frequency selective slow fading channel with the average SNR of 15 dB. Given d the users' distance from the wireless AP in km and f the carrier frequency of 2 GHz, the path loss can be written as [95],

$$PL = 40 \log_{10} d + 30 \log_{10} f + 49. \tag{5.50}$$

The throughput expression of TCP Reno is considered for the performance investigations of this chapter, thus B_i is substituted by Equation (2.4). The MSS of the TCP flows is set to the standard maximum transfer unit of an Ethernet network, which is 1460 B. It is further assumed that the end-to-end RTT for any of the TCP flows is a uniformly distributed random variable in the range [10 ms, 200 ms]. Moreover, the presented results are from 150 Monte Carlo simulations.

5.7.3 Downlink Resource Allocation Schemes (P1 and P3)

In order to solve problem (P1), first of all subcarriers are assigned based on Algorithm 3 using the assumption of equal power distribution among the allocated subcarriers. Afterwards, the optimal power distribution is calculated using the TOMLAB optimisation toolbox to solve (P1′). Similar arrangements are made for problem (P2), to allocate the set of subcarriers based on Algorithm 4, and compute the optimal power distribution by solving (P2′). First, a primary observation on the resource allocation problem (P1) is performed in a two-user scenario. Second, the process of selecting the optimal value for μ in the optimisation problem (P2) is discussed. Finally, using this value of μ, simulations are performed to examine the throughput and fairness as achieved by the two downlink resource allocation schemes.

For the benchmarks, two resource allocation problems are considered: the pure sum rate maximisation problem, denoted by (BM1), and the sum rate maximisation with an equal rate constraint, denoted by (BM2). Clearly, power and subcarrier constraints also apply to the benchmark problems. These two benchmarks represent the two extremes of the resource allocation schemes; (BM1) does not consider fairness in the allocations and aims only to achieve the maximum capacity on the link. On the other hand, (BM2) blindly provides fairness with equal rate allocation to all users.

5.7.3.1 Primary observations on the resource allocation (P1)

In order to depict a clearer picture of how problem (P1) distributes resources among users, a two-user scenario is simulated here. Figure 5.4 presents the results of this scenario as the ratio of two flows' TCP steady state throughput (B_1/B_2) on the x-axis, and FI as achieved by solving resource allocation problems (P1), and (BM2) on the y-axis. It can be seen that when the theoretical throughput of the two TCP flows are close to each other (in other words when the ratio B_1/B_2 is close to one),

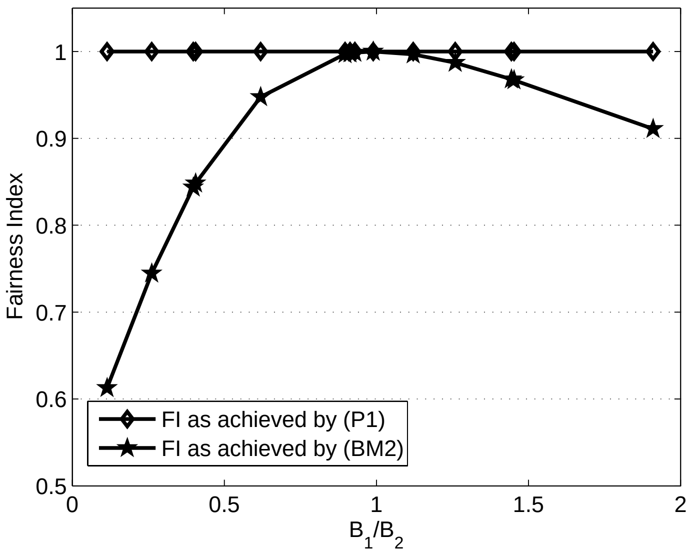


Figure 5.4: Fairness Index as achieved by solving resource allocation problems (P1), and (BM2) in two-user scenario versus the proportion of two TCP flows' theoretical throughput.

the FI achieved by both problems are almost identical. On the other hand, as B_1/B_2 diverges from one, the fairness index achieved by the proposed allocation method (i.e., problem P1) shows significant improvement. Therefore, increasing the diversity of B_i among TCP flows, can result in achieving more significant enhancements by (P1). The values of B_i for different flows departs from each other by having various RTTs, PERs or different TCP flavours.

5.7.3.2 Investigations on the selection of μ for the resource allocation scheme (P2)

The selection of μ in resource allocation problem (P2) is investigated in this section. The two simulation scenarios of two-user and ten-user are studied such that, in both the scenarios multiplier μ, which balances the two objectives, is assumed to take values from zero to two (i.e. 0, 0.3, 0.5, 0.7, 1, 1.5, and 2). Note that given μ zero, converts the resource allocation problem (P2) to the pure rate maximisation problem.

Figure 5.5 presents the achieved aggregated data rate of two users on the blue (dark

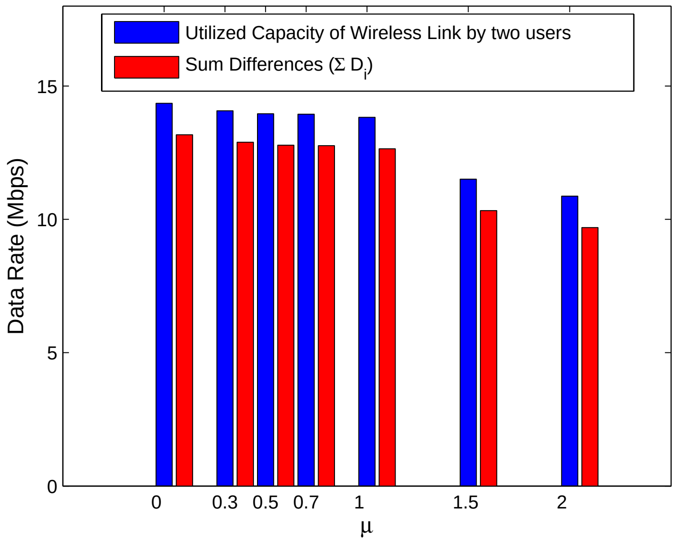


Figure 5.5: Two-User Scenario (Problem P2): Total differences between the average TCP throughput and the actual data rate on wireless link as achieved by resource allocation problem (P2) versus μ, shown on the red (light grey) bar; utilised capacity of wireless link as achieved by resource allocation problem (P2) versus μ, shown on the blue (dark grey) bar.

grey) bar, and the differences between the allocated rate and the average TCP throughput on the red (light grey) bar at various values of μ over the 150 simulation rounds. It can be seen that by increasing the value of multiplier μ from 0 to 2, the aggregated achieved data rate on wireless link is decreased, while at the same time the achieved data rate gets closer to its optimal value from the TCP perspective.

In the second scenario number of mobile users is increased to ten, with the same configuration to scenario one. Figure 5.6 shows the total differences between the achieved data rate by each TCP flow and its corresponding theoretical throughput on the red (light grey) bar, and the achieved aggregated data rate on the wireless link on the blue (dark grey) bar for various values of multiplier μ in this scenario. Similar observation to scenario one can be seen in Figure 5.6. Across the range of values for multiplier μ, the utilised capacity of wireless network is decreased by 14%, although the overall achieved data rate is 20% closer to the average end-to-end capacity, which is defined by the theoretical TCP throughput.

The effect of resource allocation problem (P2) on the level of fairness among com-

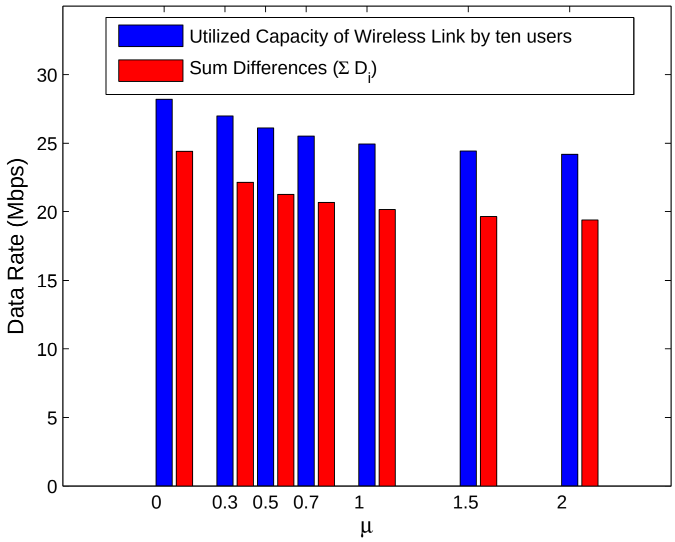


Figure 5.6: Ten-User Scenario (Problem P2): Total differences between the average TCP throughput and actual data rate on wireless link as achieved by resource allocation problem (P2) versus μ, shown on the red (light grey) bar; utilised capacity of wireless link as achieved by resource allocation problem (P2) versus μ, shown on the blue (dark grey) bar.

peting TCP flows is also investigated having the similar set of values for μ. The introduced index of fairness in section 5.7.1, FI, is calculated for these simulated scenarios, and results can be seen in Figure 5.7. Observing from this figure, when multiplier μ takes the value of 0.3, the fairness index has its maximum improvements, although across various values of μ, fairness among TCP flows is increased comparing with the benchmark problem (μ=0). Hence, from the studied simulation scenarios, the recommended value of balancing two objectives in the optimisation problem (P2) is 0.3, which improves the overall performance i.e. sum rate versus fairness, more significantly. Thus, the performance of problem (P2) with μ equal to 0.3, is further investigated in the next scenarios.

5.7.3.3 Investigations on the performance of (P1 and P2)

At this stage, the performance of resource allocation problems (P1) and (P2) with μ equal to 0.3 is investigated. The aggregated throughput on the downlink and the level of fairness as achieved by the two TCP-aware resource allocation problems (P1

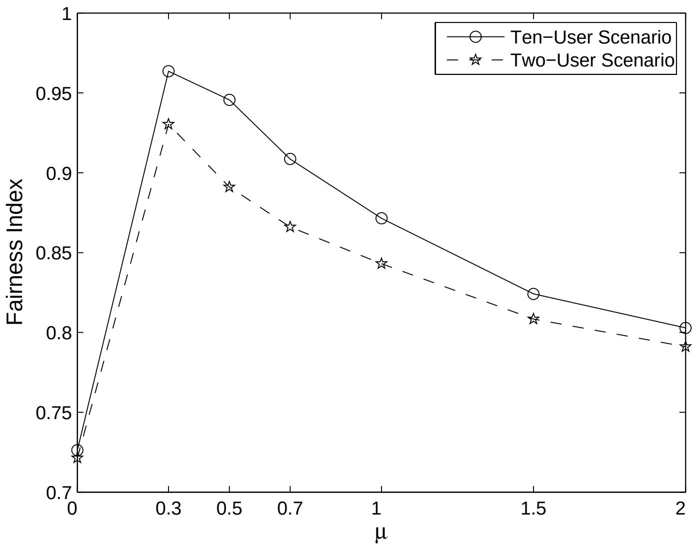


Figure 5.7: Ten-User Scenario(Problem P2): Mean Value of fairness index as achieved by resource allocation problem (P2) versus μ, shown on the left axis; utilised capacity of wireless link as achieved by resource allocation problem (P2) versus μ, shown on the right axis.

and P2) and the two benchmarks (BM1 and BM2) are computed and compared. Simulations are performed over various number of mobile users from 2 to 20, and the results are presented in Figures 5.8 and 5.9.

Observe from Figure 5.8 that, the average increase in the fairness index is 40% from (BM1) to (P1), and 20% from (BM2) to (P1). In the scenario with larger number of mobile users competing over the wireless channel, providing the fair distribution for the benchmark resource allocation schemes is more challenging. Thereby, the proposed TCP-aware schemes enhance the fairness index more significantly among larger number of users, e.g. this index is increased up to 65% in twenty-user scenario. Despite the increase in fairness among TCP flows, the aggregated data rate on wireless link is decreased. This effect can be seen in Figure 5.9, in which the aggregated throughput shows a decrement of 10% and 5% in average comparing the results of problem (P1) with (BM1) and (BM2) successively. The achieved sum data rate plotted in Figure 5.9, is based on the allocated data rate in wireless link which can be utilised by the user regarding its end-to-end capacity. Further observation shows that, maximum degradation in the sum rate is also in the twenty-user scenario,

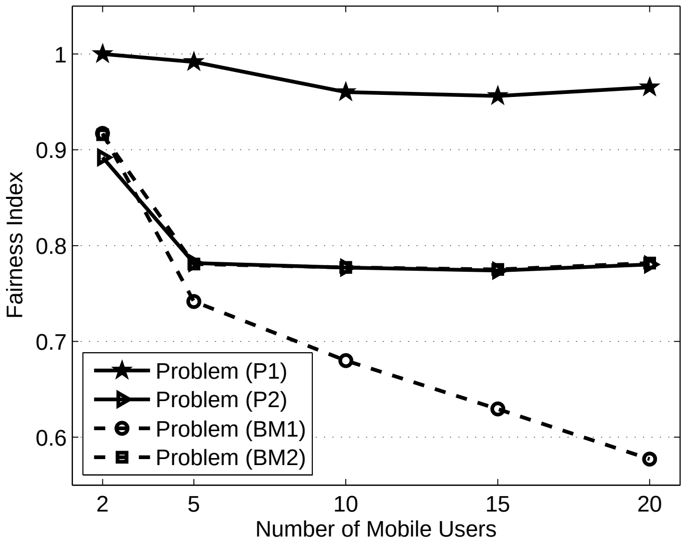


Figure 5.8: Fairness Index as achieved by solving resource allocation problems (P1), (P2), (BM1), and and (BM2) versus the number of mobile users.

comparing the proposed TCP-aware scheme (P1) with the pure rate maximisation (BM1) that is 12.5% decrease in the overall throughput.

5.7.4 Joint Uplink-Downlink Resource Allocation Schemes (P3 and P4)

In order to examine the performance of the joint uplink-downlink resource allocation problems (P3 or P4), the optimal value of m_d should be computed. Thus, the iterations in Algorithm 5 (or Algorithm 6) are used to find this value. Simulations are performed with the optimal value of m_d, the number of downlink subcarriers, where it can be seen that in addition to the previously discussed enhancements in the fairness among downlink TCP flows, the aggregated end-to-end downlink throughput is increased. The increase in the aggregated downlink throughput has two main reasons. First, in the situations where there is light traffic on the uplink and extra resources are available, this extra resources are allocated to the downlink. Second, and more important from the perspective of this research work, in the situations where the uplink is busy and there are not sufficient resources for delivery

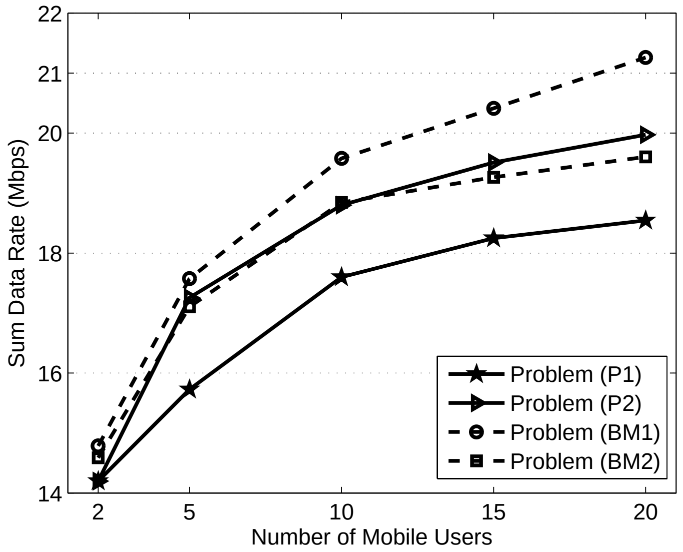


Figure 5.9: Sum data rate (Mbps) as achieved by solving resource allocation problems (P1), (P2), (BM1), and and (BM2) versus the number of mobile users.

of the ACK packets in response to the allocated downlink data rate, increasing the uplink can guarantee TCP flows on the downlink to accomplish their allocated data rate.

5.7.4.1 Investigations on the performance of (P3 and P4)

In this simulation scenario, m_d is initialised with 32 subcarriers, thus the remaining 20 subcarriers are allocated to uplink. Afterwards step (f) in Algorithm 5 (or 6), finds the optimal value of m_d in few iterations. The benchmark problems are similar to the problems used for the performance investigation of the downlink schemes. Thus, benchmark problems (BM1) and (BM2) operate at the $m_d = 32$.

As discussed earlier, under the conditions where the available resources in the uplink are more than required to be allocated for uplink traffic, these wireless resources can be allocated to downlink and increase the downlink throughput. On the other hand, when the available resources for the uplink channel can not satisfy the data rate requirements of the uplink, increasing the number of uplink subcarriers guarantee the delivery of the ACK packets, thus enhancing the achievable throughput on the

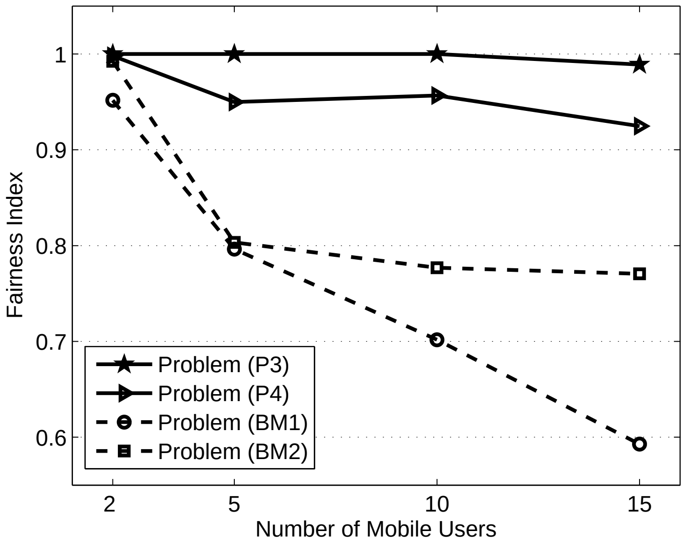


Figure 5.10: Fairness Index as achieved by solving resource allocation problems (P3), (P4), (BM1), and (BM2) versus the number of mobile users.

downlink.

The results from this simulation scenario are presented in Figures 5.10-5.12, in which the number of users is increased from 2 to 20. Observed from Figure 5.10, it can be seen that the level of enhancement in the fairness index is significant. Moreover, Figures 5.12 and 5.11 show that dynamic allocation of the border between uplink and downlink improve the total aggregated throughput and the aggregated downlink throughput by approximately 15%. Therefore, setting m_d dynamically and in accordance with the requirements of TCP connection, enhances the total achieved throughput.

5.8 Concluding Remarks

In this chapter, a framework for TCP-aware resource allocation scheme over OFDMA wireless networks is detailed. Different formulations of such a problem are presented and the achieved performance is investigated. Although various research works studied the problem of subcarrier/power allocation in OFDMA, aspects such as those

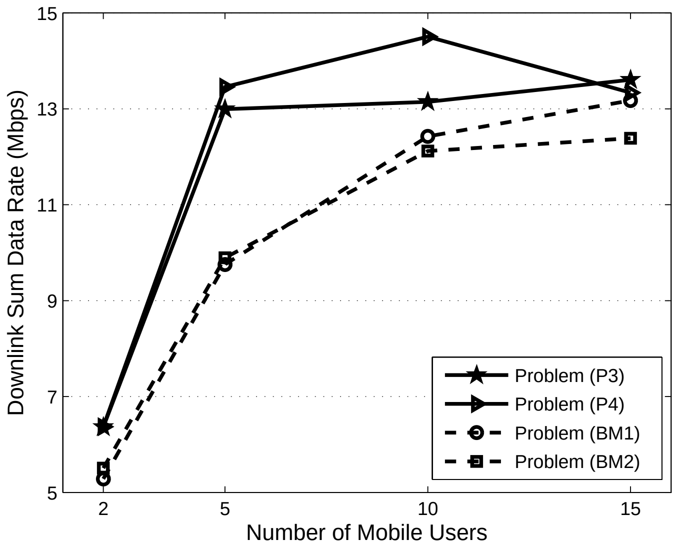


Figure 5.11: Downlink sum data rate (Mbps) as achieved by solving resource allocation problems (P3), (P4), (BM1), and (BM2) versus the number of mobile users.

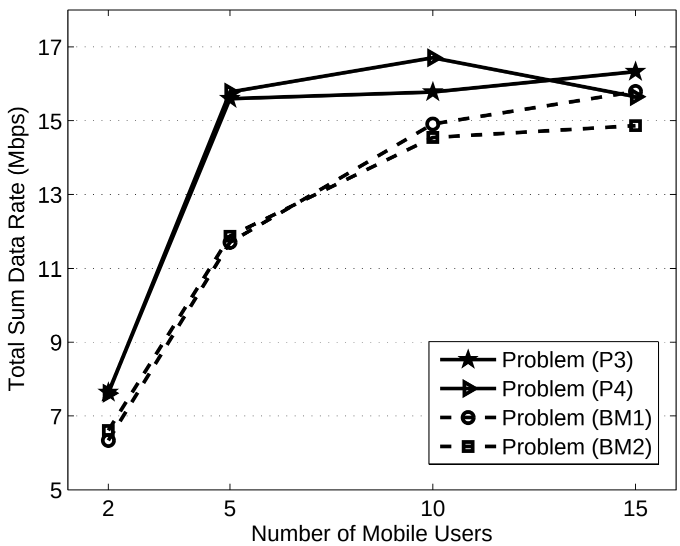


Figure 5.12: Total (Uplink+Downlink) sum data rate (Mbps) as achieved by solving resource allocation problems (P3), (P4), (BM1), and (BM2) versus the number of mobile users.

pertaining to the end-to-end data transmission perspective have not been sufficiently addressed. Therefore, in this work, the theoretical TCP throughput, which is the actual throughput that the end-to-end flow can achieve, is used as a means of providing TCP-awareness. Simulation studies reveal that the proposed TCP-aware resource allocation scheme over downlink OFDMA can enhance fairness among TCP competing flows, and achieve a more balanced data rate towards the TCP throughput.

The problem of scarce and limited availability of resources on the uplink is studied and its effect on the performance of TCP over the downlink is examined. Thereby, the joint uplink-downlink resource allocation problem is proposed to address this issue, and guarantee the delivery of allocated data rate on the downlink, by assigning sufficient resources to the uplink. The rational of this joint problem is the bi-directional nature of TCP connections, and it is mainly motivated by the increasing amount of the uplink traffic in the network.

Algorithm 5 Subcarrier Allocation to the Optimisation Problem (P3)

a) Initialisation

1. $m_d = \frac{1}{1+\rho} \cdot m$.
2. Set R_{d_i}=0 and $\Omega_i = \phi$ for i=1 to n and $C_d = \{1, 2, ...m_d\}$.
3. Set R_{u_i}=0 and $\Psi_i = \phi$ for i=1 to n and $C_u = \{m_d + 1, ...m\}$.
4. Sort the users' index in the descending order of B_i.

b) for i=1 to n

1. Find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C_d$.
2. Let $\Omega_i = \Omega_i \cup \{k\}$ and $C_d = C_d - \{k\}$.
3. Update R_{d_i}

c) while $C_d \neq \phi$

1. Find user l satisfying $R_{d_l}/B_l < R_{d_i}/B_i$ for all $i \in \{1, ..., n\}$.
2. For user l, find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C_d$.
3. Let $\Omega_l = \Omega_l \cup \{k\}$ and $C_d = C_d - \{k\}$.
4. Update R_{d_i}

d) for i=1 to n

1. Find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C_u$.
2. Let $\Psi_i = \Psi_i \cup \{k\}$ and $C_u = C_u - \{k\}$.
3. Update R_{u_i}

e) while $C_u \neq \phi$

1. Find user l satisfying $(R_{u_l} - \rho R_{d_l}) < (R_{u_i} - \rho R_{d_i})$ for all $i \in \{1, ..., n\}$.
2. For user l, find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C_u$.
3. Let $\Psi_l = \Psi_l \cup \{k\}$ and $C_u = C_u - \{k\}$.
4. Update R_{u_i}

f) Check if constraint (5.21) is satisfied.

1. Yes: $m_d \leftarrow m_d - 1$.
2. No: $m_d \leftarrow m_d + 1$.

g) Go to a(2).

Algorithm 6 Subcarrier Allocation to the Optimisation Problem (P4)

a) Initialisation

1. $m_d = \frac{1}{1+\rho} \cdot m$.
2. Set R_{d_i}=0 and $\Omega_i = \phi$ for i=1 to n and $C_d = \{1, 2, ...m_d\}$.
3. Set R_{u_i}=0 and $\Psi_i = \phi$ for i=1 to n and $C_u = \{m_d + 1, ...m\}$.
4. Sort the users' index in the descending order of B_i.

b) for i=1 to n

1. Find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C_d$.
2. Let $\Omega_i = \Omega_i \cup \{k\}$ and $C_d = C_d - \{k\}$.
3. Update R_{d_i}

c) while $C_d \neq \phi$

1. Find user l satisfying $R_{d_l} - \mu D_l < R_{d_i} - \mu D_i$ for all $i \in \{1, ..., n\}$.
2. For user l, find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C_d$.
3. Let $\Omega_l = \Omega_l \cup \{k\}$ and $C_d = C_d - \{k\}$.
4. Update R_{d_i}

d) for i=1 to n

1. Find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C_u$.
2. Let $\Psi_i = \Psi_i \cup \{k\}$ and $C_u = C_u - \{k\}$.
3. Update R_{u_i}

e) while $C_u \neq \phi$

1. Find user l satisfying $(R_{u_l} - \rho R_{d_l}) < (R_{u_i} - \rho R_{d_i})$ for all $i \in \{1, ..., n\}$.
2. For user l, find the subcarrier k satisfying $|G_{ik}| > |G_{ij}|$ for all $j \in C_u$.
3. Let $\Psi_l = \Psi_l \cup \{k\}$ and $C_u = C_u - \{k\}$.
4. Update R_{u_i}

f) Check if constraint (5.29) is satisfied.

1. Yes: $m_d \leftarrow m_d - 1$.
2. No: $m_d \leftarrow m_d + 1$.

g) Go to a(2).

Chapter 6

Conclusions and Future Research

6.1 Concluding Remarks

This thesis has discussed my views in the design of a smart TCP-aware link-layer. The objective of this design is to improve the performance of the end-to-end TCP flows, using algorithms that are compatible with the existing TCP/IP protocol stack so that no modifications will be required at the transport protocol. To this end, three TCP-aware techniques are proposed: TCP-aware ARQ mechanism, TCP-aware FEC rate selection, and TCP-aware wireless resource allocation scheme. By extensive simulations, performance of the proposed schemes has been investigated. Various figures of merit such as the end-to-end throughput, and fairness among TCP flows, have been explored to present the efficiency of the proposed algorithms in wireless networks.

The first proposed technique aims to adapt the FEC code rate prior to packet transmission so that fairness among heterogeneous TCP flows is accomplished. Utilising information on the TCP flavour of each flow, a framework has been detailed to maximise fairness among these flows with respect to Jain's fairness index. The real-time solution of the proposed problem has been discussed using a heuristic approach, while numerical observations have confirmed its convergence to the optimum value.

Thorough simulation studies using the OPNET modeler, under various packet loss probabilities and RTT conditions, have revealed that fairness as achieved by the proposed scheme increases significantly compared to the channel based rate adaptation for FEC.

Second, a TCP-aware ARQ mechanism has been proposed that assigns its persistency dynamically on a per packet basis. Using the timing information of TCP such that RTT packets are prioritised in the retransmission queue so that timer expiry at the transport layer could be minimised. Moreover, the expired packets at the transport layer are dropped from the ARQ retransmission queue in order to avoid extra retransmissions. Simulation investigations using OPNET modeler have shown that implementing the TCP-aware dynamic ARQ mechanism could significantly increase the end-to-end performance of TCP. In addition, by deploying packet drop due to expiry at the transport layer, the number of retransmissions over the wireless link are decreased, thus the wireless link is utilised more efficiently. Moreover, the queuing analysis of the proposed scheme has shown that the resultant queuing delay is unchanged.

Finally, the problem of subcarrier/power allocation in OFDMA has been investigated. The TCP-aware resource allocation algorithm has been proposed which contributes to the performance of the end-to-end data transmissions in two ways. First, to provide fairer throughput towards the achievable throughput by TCP in the downlink, the theoretical TCP throughput has been added to the constraints of the downlink resource allocation problem. Second, to address the problem of asymmetric links and the effect of available uplink resources on the downlink performance, a joint uplink-downlink resource allocation scheme has been proposed.

To this end, two different formulations of the TCP-aware downlink resource allocation problem have been presented. In the first problem, a set of non-linear constraints are added, to maintain the proportional downlink rate among users with respect to the TCP theoretical throughput. The second problem attempts to minimise the gap between the allocated data rate and the theoretical TCP throughput.

A wide range of simulation scenarios have been carried out to investigate the effect of these resource allocation schemes on the performance of the end-to-end TCP flows. The simulation results have revealed that not only is more balanced throughput towards TCP throughput achieved but also fairness among downlink TCP flows is improved significantly.

The second part of this problem has addressed the issue of scarce availability of resources on the uplink that could result in the degradation of the downlink throughput. This issue and its effect on the performance of TCP has been addressed by proposing a joint uplink-downlink resource allocation scheme that performs in a TCP-aware fashion. The proposed scheme has constrained the minimum uplink data rate of each TCP flow based on its allocated data rate in the downlink. The above mentioned constraint is because of the bi-directional nature of TCP connections, which require sufficient bandwidth in the uplink in order to guarantee the delivery of the downlink packets. The performance of this joint uplink-downlink resource allocation problem has been investigated with a wide range of simulation scenarios. It has been shown that by using the proposed resource allocation scheme, the aggregated end-to-end throughput is increased significantly.

In addition to the asymmetric links, a number of other problems can be addressed using the TCP-aware resource allocation scheme. For example, a large number of TCP connections are short-lived connections, and how they perform is highly dependent on their performance at the slow start phase of TCP. Therefore, by providing a closer wireless data rate to the TCP throughput, the TCP-aware resource allocation scheme can help the short-lived TCP flows to perform well. Moreover, this resource allocation scheme provides an opportunity for the congestion window of the newly established TCP connections to grow and capture the available wireless channel accordingly [96].

6.2 Avenues of Future Research

In this section, I would like to open the following interesting issues that among many others can be continued by the research presented in the thesis.

- **Co-Existence of Multiple TCP Flavours**
  Numerous variants of TCP have been proposed in the past few years among which many are being used over the current Internet [64]. To this end, co-existence of multiple flavours of TCP has been studied in Chapter 3 of this thesis, where the adaptive FEC scheme was proposed. This issue can be widely studied within the techniques such as the TCP-aware resource allocation schemes. More specifically, in the joint uplink-downlink allocation, various TCP flavours may have different constraints on the uplink, e.g., delay-based congestion control algorithms are penalised more significantly by the scarcity of resources on the uplink comparing with loss-based congestion controls [19]. Moreover, some variants of TCP such as using the SACK option, which uses the transmission of larger ACK packets, thus requires larger bandwidth over the uplink. On the other hand, some variants of TCP are more tolerant to the packet loss, e.g., TCP Westwood. These issues can be considered in the resource allocation schemes so that implicitly incorporating the weighting parameters in the constraints (5.21) or (5.29). These parameters are based on the flavour of TCP at the end-host of the corresponding TCP flow.

- **Application Awareness**
  Motivated by the fact that a plethora of different applications, ranging from streaming video to file transferring are using TCP, application awareness can be added in the TCP-aware resource allocation scheme. The rationale behind this idea is that different applications have various requirements. For example, applications such as VoIP and streaming require ramping up to a specific data rate quickly. Elastic applications such as Email and HTTP grow smoother and more constantly [97]. Thus, a set of utility functions that capture the requirements of these applications can be defined so that optimising the utili-

ties results in the optimal resource allocation scheme. A preliminary study of such a resource allocation scheme has been performed in [98], which shows the enhancements in the end-to-end performance of TCP.

- **Modifications to the TCP**

  Combining the design of the smart link-layer with modifications to the actual TCP congestion control mechanisms can potentially result in a significant performance improvement. Although dismantling the whole engineering design of congestion control may not lead to a stable design, minimal changes can benefit the overall efficiency. Inspired by the proposal of Compound TCP (CTCP) [20] that combines the loss-based and delay-based congestion controls, introducing new set of parameters to capture the effect of wireless packet transmission can enhance TCP performance without degrading its performance or increasing the overhead over a range of other networks. Using such concepts, it may be possible to answer open-ended questions such as, how can channel quality information be used by the transport layer protocols to more efficiently utilise wireless resources in a more end-to-end manner.

References

[1] V. Jacobson, "Congestion Avoidance and Control," *Proc. ACM SIGCOMM '88*, Stanford, USA, Aug. 1988. 20, 30, 32

[2] W. John and S. Tafvelin, "Analysis of Internet Backbone Traffic and Header Anomalies observed," *Proc. ACM SIGCOMM*, pp. 111–116, Kyoto, Japan, Aug. 2007. 21, 28

[3] A. S. Tanenbaum, *Computer Networks.* Prentice Hall, 2003. 28

[4] Information Science Institute, University of Southern California, CA, USA, "Transmission Control Protocol," *IETF RFC 793*, Sep. 1981. 28

[5] S. Ebrahimi-Taghizadeh, A. Helmy, and S. Gupta, "TCP vs. TCP: a Systematic Study of Adverse Impact of Short-lived TCP Flows on Long-lived TCP Flows," *Proc. IEEE INFOCOM'05*, Miami, USA, Mar. 2005. 28

[6] V. Paxson and M. Allman, "Computing TCP's Retransmission Timer," *IETF RFC 2988*, Nov. 2000. 30, 85

[7] C. Barakat, E. Altman, and W. Dabbous, "On TCP performance in a heterogeneous network: a survey," *IEEE Commun.*, vol. 38, pp. 40–46, 2000. 32

[8] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," *IETF RFC 2581*, Apr. 1999. 33

[9] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," *IETF RFC 2001*, Jan. 1997. 33, 57, 80

[10] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm," *IETF RFC 3782*, Apr. 2004. 33, 57

[11] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP," *ACM Comp. Commun. review*, vol. 26, pp. 5–21, Jul. 1996. 34

[12] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment Options," *IETF RFC 2018*, Oct. 1996. 34, 57

[13] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP," *IETF RFC 2883*, Jul. 2000. 34

[14] M. Welzl, "A Survey of Lower-than-Best Effort Transport Protocols," *IETF Internet Draft*, Mar. 2009. 34

[15] L. L. P. Lawrence S. Brakmo, Sean W. O'Malley, "TCP Vegas: New Techniques for Congestion Detection and Avoidance," *Proc. ACM SIGCOMM '94*, vol. 24, pp. 24–35, London, UK, Oct. 1994. 34

[16] R. Wang, K. Yamada, M. Y. Sanadidi, and M. Gerla, "TCP with Sender-side Intelligence to handle Dynamic, Large, Leaky Pipes," *IEEE J Sel. Areas Commun.*, vol. 23, pp. 235–248, Mar. 2005. 35, 46, 57

[17] S. Floyd, "HighSpeed TCP for Large Congestion Windows," *IETF RCF 3649*, Dec. 2003. 36

[18] T. Kelly, "Scalable TCP: Improving Performance in HighSpeed Wide Area Networks," *Proc. International Workshop on Protocols for Fast Long Distance Networks*, Geneva, Switzerland, Dec. 2003. 36

[19] C. Jin, D. Wei, and S. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance," *Proc. IEEE INFOCOM'04*, Hong Kong, Mar. 2004. 36, 133

[20] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A Compound TCP Approach for High-speed and Long Distance Networks," *Proc. INFOCOM'06*, Barcelona, Spain, Apr. 2006. 37, 134

[21] H. Balakrishnan, V. N. Padmanabhan, G. Fairhurst, and M. Sooriyabandara, "TCP Performance Implications of Network Path Asymmetry," *IETF RFC 3449*, Dec. 2002. 37, 96, 100

[22] S. Floys, A. Arcia, and J. Iyengar, "Adding Acknowledgement Congestion Control to TCP," *IETF Internet Draft*, Jan. 2009. 37, 96, 100

[23] J. Postel, "User Datagram Protocol," *IETF RFC 768*, Aug. 1980. 38

[24] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," *IETF RFC 3448*, Jan. 2003. 38

[25] E. Kohler, M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," *IETF RFC 4340*, Mar. 2006. 39

[26] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, "Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation," *IEEE/ACM Trans. Net.*, vol. 8, pp. 133–145, Apr. 2000. 40, 62

[27] Z. Chen, T. Bu, M. Ammar, and D. F. Towsley, "Comments on Modeling TCP Reno Performance: A simple model and its Empirical Validation," *IEEE/ACM Trans. Net.*, vol. 14, pp. 451–453, Apr. 2006. 40, 62

[28] B. Sikdar, S. Kalyanaraman, and K. S. Vastola, "Analytic Models for the Latency and Steady-State Throughput of TCP Tahoe, Reno, and SACK," *IEEE/ACM Trans. Net.*, vol. 11, pp. 959–971, Dec. 2003. 40, 62

[29] R. Dunaytsev, Y. Koucheryavy, and J. Harju, "TCP NewReno Throughput in the Presence of Correlated Losses: The Slow-but-Steady Variant," *Proc. IEEE INFOCOM'06, Global Internet Workshop*, Barcelona, Spain, Apr. 2006. 41, 62

[30] A. Zanella, G. Procissi, M. Gerla, and M. Y. Sanadidi, "TCP Westwood: Analytic Model and Performance Evaluation," *Proc. IEEE GLOBECOM'01*, Texas, Dec. 2001. 41, 62

[31] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update," *White Paper*, Jan. 2009. 42

[32] "IEEE Std. 802.11-1997: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," *IEEE Computer Society*, Jun. 1997. 43

[33] "IEEE Std. 802.16-2004: Air Interface for Fixed Broadband Wireless Access Systems," *IEEE Computer Society and IEEE Microwave Theory and Techniques Society*, Oct. 2004. 44

[34] "IEEE Std. 802.16a-2003: Air Interface for Fixed Broadband Wireless Access Systems, Medium Access Control Modifications and Additional Physical Layer Specifications for 2-11 GHz," *IEEE Computer Society and IEEE Microwave Theory and Techniques Society*, Apr. 2003. 44

[35] "IEEE Std. 802.16e-2005: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands," *IEEE Computer Society and IEEE Microwave Theory and Techniques Society*, Feb. 2006. 44, 98

[36] TR 36.913, V8.0.0, "Requirements for Further Advancements for E-UTRA (LTE-Advanced)," *3rd Generation Partnership Project, Technical Specification Group Radio Access Network*, Jun. 2008. 45

[37] V. Tsaoussidis and I. Matta, "Open Issues on TCP for Mobile Computing," *Wireless Commun. and Mobile Comput.*, vol. 2, pp. 3–20, Feb. 2002. 46

[38] H. Elaarag, "Improving TCP Performance over Mobile Networks," *ACM Computing Suryeys*, vol. 34, pp. 357–374, Sep. 2002. 46

[39] X. Chen, H. Zhai, J. Wang, and Y. Fang, "A survey on improving TCP performance over wireless networks," *Springer Net. Theory & Appl. Series*, vol. 16, pp. 657–695, 2005. 46, 95

[40] Y. Tian, K. Xu, and N. Ansari, "TCP in Wireless Environments: Problems and Solutions," *IEEE Commun.*, vol. 43, pp. S27–S32, Mar. 2005. 46, 57

[41] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN)," *IETF RFC 3168*, Sep. 2001. 46

[42] S. Biaz and N. H. Vaidya, "De-Randomizing Congestion Losses to Improve TCP Performance Over Wired-Wireless Networks," *IEEE/ACM Trans. Net.*, vol. 13, pp. 596–608, Jun. 2005. 46, 47

[43] A. Bakre and V. Paxon, "I-TCP: Indirect TCP for mobile hosts," *Proc.15th International Conference on Distributed Computing System*, pp. 136–143, Jun. 1995. 47

[44] K. Brown and S. Singh, "M-TCP: TCP for Mobile Cellular Networks," *Proc. ACM SIGCOMM'97*, Cannes, France, Sept. 1997. 47

[45] W. Wei, C. Zhang, H. Zang, J. Kurose, and D. Towsley, "Inference and Evaluation of Split-Connection Approaches in Cellular Data Networks," *Proc. Passive and Active Measurement (PAM)*, 2006. 48

[46] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving TCP/IP Performance over Wireless Networks," *Proc. ACM MOBICOM '95*, Berkeley, USA, Nov. 1995. 49

[47] P. Sarolahti and S. Floyd, "Cross-layer Indications for Transport Protocols," *IETF Internet Draft*, Oct. 2006. 49

[48] G. Fairhurst and L. Wood, "Advice to Link Designers on Link Automatic Repeat reQuest (ARQ)," *IETF RFC 3366*, Aug. 2002. 50, 82, 83

[49] F. Vacirca, A. D. Vendictis, A. Todini, and A. Baiocchi, “On the Effects of ARQ Mechanisms on TCP Performance in Wireless Environments,” *Proc. IEEE GLOBECOM’03*, San Fransisco, USA, Dec. 2003. 51, 83

[50] A. L. Toledo and X. Wang, “TCP Performance over Wireless MIMO Channels with ARQ and Packet Combining,” *IEEE Trans. Mobile Comput.*, vol. 5, pp. 208–223, Mar. 2006. 52

[51] D. Kliazovich, F. Granelli, and M. Gerla, “Performance Improvement in Wireless Networks using Cross-Layer ARQ,” *International J Comp. and Telecommun. Net.*, vol. 51, pp. 4396–4411, Oct. 2007. 52

[52] C. Barakat and E. Altman, “Bandwidth tradeoff between TCP and Link-Level FEC,” *Comp. Net.*, vol. 39, pp. 133–150, June 2002. 53, 59, 60

[53] D. Barman, I. Matta, E. Altman, and R. Azouzi, “TCP Optimization through FEC, ARQ and Transmission Power Tradeoffs,” *Proc. 2nd Int’l Conf. on Wired/Wireless Internet Commun.*, Frankfurt/Oder, Germany, Feb. 2004. 53

[54] R. Abdelmoumen, M. Malli, and C. Barakat, “Analysis of TCP Latency over Wireless Links supporting FEC/ARQ-SR for Error Recovery,” *Proc. IEEE ICC’04*, Paris, France, Jun. 2004. 53

[55] J. P. Singh, Y. Li, N. Bambos, A. Bahai, B. Xu, , and G. Zimmermann, “TCP Performance Dynamics and Link-Layer Adaptation Based Optimization Methods for Wireless Networks,” *IEEE Trans. Wireless Commun.*, vol. 6, pp. 1864–1879, May 2007. 53

[56] H. Jiang, W. Zhuang, and X. Shen, “Cross-Layer Design for Resource Allocation in 3G Wireless Networks and Beyond,” *IEEE Commun.*, vol. 43, pp. 120–126, Dec. 2005. 54, 99

[57] M. Ghaderi, A. Sridharan, H. Zang, D. Towsley, and R. Cruz, “TCP-Aware Channel Allocation in CDMA Networks,” *IEEE Trans. Mobile Comput.*, vol. 8, pp. 14–28, Jan. 2009. 54

[58] M. Chiang, "Balancing Transport and Physical Layers in Wireless Multihop Networks: Jointly Optimal Congestion Control and Power Control," *IEEE J Sel. Areas Commun.*, vol. 23, pp. 104–116, Jan. 2005. 54, 99

[59] S. Kim and I. Yeom, "TCP-aware Uplink Scheduling for IEEE 802.16," *IEEE Commun. Letters*, vol. 11, pp. 146–148, Feb. 2007. 54, 99

[60] W. R. Stevens, *TCP/IP illustrated, Volume I The protocols.* Addison Wesley, Feb. 2000. 55, 89

[61] B. Veal, K. Li, and D. Lowenthal, "New Methods for Passive Estimation of TCP Round Trip Times," *ACM Comp. Commun. Review*, vol. 3431, pp. 121–134, 2005. 55

[62] H. Jiang and C. Dovrolis, "Passive Estimation of TCP Round-Trip Times," *ACM Comp. Commun. Review*, vol. 32, pp. 75–88, July 2002. 56

[63] S. Jaiswal, *Measurement in the Middle: Inferring End-End Path Properties and Characteristics of TCP Connections through Passive Measurement.* PhD thesis, University of Massachusetts Amherst, Sep. 2005. 56

[64] A. Medina, M. Allman, and S. Floyd, "Measuring the Evolution of Transport Protocols in the Internet," *ACM Comp. Commun. Review*, vol. 35, pp. 37–52, Apr. 2005. 57, 133

[65] T. Mahmoodi, O. Holland, V. Friderikos, and H. Aghvami, "Cross-Layer Optimization of the Link-Layer based on the Detected TCP Flavor," *WWRF 20th*, Ottawa, Canada, May 2008. 58

[66] T. Mahmoodi, O. Holland, V. Friderikos, and H. Aghvami, "Cross-Layer Optimization of the Link-Layer based on the Detected TCP Flavor," *proc. IEEE PIMRC'08*, Cannes, France, Sep. 2008. 58

[67] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems," Tech. Rep. 301, Sep. 1984. 58, 61, 64, 116

[68] T. Mahmoodi, V. Friderikos, O. Holland, and A. H. Aghvami, "Cross-layer Optimization to Maximize Fairness among TCP Flows of different TCP Flavors," *Proc. IEEE GLOBECOM'08*, New Orleans, USA, Nov. 2008. 58

[69] T. Mahmoodi, V. Friderikos, O. Holland, and A. H. Aghvami, "Optimal Design of FEC for Fairness Maximisation among TCP flavours over Wireless," *submitted to IET Commun.*, 2010. 58

[70] F. Vacirca, A. De Vebductis, and Baiocchi, "Optimal Design of Hybrid FEC/ARQ Schemes for TCP over Wireless Links with Rayleigh Fading," *IEEE Trans. Mobile Comput.*, vol. 5, pp. 289–302, Apr. 2006. 59, 60

[71] J. G. Proakis, *Digital Communications.* McGraw Hill, 3rd ed., 1995. 60, 65

[72] S. Pilosof, R. Ramjee, Y. Shavitt, and P. Sinha, "Understanding TCP fairness over Wireless LAN," *Proc. IEEE INFOCOM'03*, San Fransisco, USA, Mar. 2003. 61

[73] N. Blefari-Melazzi, A. Dett, I. Habib, A. Ordine, and S. Salsano, "TCP Fairness Issues in IEEE 802.11 Networks: Problem Analysis and Solutions Based on Rate Control," *IEEE Trans. Wireless Commun.*, vol. 6, pp. 1346–1355, Apr. 2007. 61

[74] E. Lopez-Aguilera, M. Heusse, F. Rousseau, A. Duda, and J. Casademont, "Performance of Wireless LAN Access Methods in Multicell Environments," *Proc. IEEE GLOBECOM'06*, San Francisco, USA, Nov. 2006. 61

[75] S. Boyds and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004. 69, 147, 148

[76] "The TCP Westwood Model for the ns2 Network Simulator," *http://www.cs.ucla.edu/nrl/hpi/tcpw.* 71

[77] C. Gorg and U. Vornefeld, "Uniform Distribution of Users in a Circle," *The 100 most frequent errors in performance evaluation*, http://www.comnets.uni-bremen.de/itg/itgfg521, Aug. 2007. 72, 92

[78] J. J. Lee and M. Gupta, "A New Traffic Model for Current User Web Browsing Behavior," *white paper, Intel Corporation*, Sep. 2007. 73

[79] 4i2i Communication Ltd., *Convolutional Encoder and Viterbi Decoder*, 3.1 ed., Dec. 2001. 72, 74

[80] D. Bertsekas and R. Gallager, *Data Networks.* Prentice Hall, 1992. 86, 88

[81] L. Kleinrock, *Queueing Systems, Volume II: Computer Applications.* John Wiley and Sons, 1976. 88

[82] TR 25.814, V7.1.0, "Physical Layer Aspects for Evloved Universal Radio Access (UTRA)," *3rd Generation Partnership Project, Technical Specification Group Radio Access Network*, Sep. 2006. 98, 104, 116

[83] I. C. Wong and B. L. Evans, "Optimal Downlink OFDMA Resource Allocation with Linear Complexity to Maximize Ergodic Rates," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 962–971, Feb. 2008. 99

[84] Z. Han, Z. Ji, and K. J. R. Liu, "Power Minimization for Multi-Cell OFDM Networks Using Distributed Non-cooperative Game Approach," *Proc. IEEE GLOBECOM'04*, Dallas, USA, Nov. 2004. 99

[85] L. Hoo, B. Halder, J. Tellado, and J. M. Cioffi, "Multiuser Transmit Optimization for Multicarrier Broadcast Channels: Asympotic FDMA capacity region and algorithms," *IEEE Trans. Commun.*, vol. 52, pp. 922–930, Jun. 2004. 99

[86] Z. Shen, J. G. Andrews, and B. L. Evans, "Adaptive Resource Allocation in Multiuser OFDM Systems With Proportional Rate Constraints," *IEEE Trans. Wireless Commun.*, vol. 4, pp. 2726–2737, Nov. 2005. 99, 109, 111, 112

[87] W. Rhee and J. M. L. Cioffi, "Increase in Capacity of Multiuser OFDM System using Dynamic Subchannel Allocation," *Proc. IEEE VTC Spring*, Tokyo, Japan, May 2000. 99

[88] Z. Han, Z. Ji, and K. J. R. Liu, "Fair Multiuser Channel Allocation for OFDMA Networks Using Nash Bargaining Solutions and Coalitions," *IEEE Trans. Commun.*, vol. 35, pp. 1366–1375, Aug. 2005. 99, 101, 111

[89] R. Iyengar, K. Kar, B. Sikdar, and X. Luo, "QoS Provisioning and Radio Resource Allocation in OFDMA based WiMAX Systems," *White paper*. 99

[90] M. Ghaderi, A. Sridharan, and H. Zang, "TCP-aware Resource Allocation in CDMA Networks," *Proc. ACM MobiCom'06*, Sep. 2006. 99

[91] S. T. Chung and A. J. Goldsmith, "Degrees of freedom in adaptive modulation: A unified view," *IEEE Trans. Commun.*, vol. 49, pp. 1561–1571, Sep. 2001. 101

[92] T. Mahmoodi, V. Friderikos, O. Holland, and A. H. Aghvami, "TCP-aware Resource Allocation Problem in OFDMA based Wireless Networks," *Proc. IWCLD*, Palma, Spain, Jun. 2009. 102

[93] H. Balakrishnan, H. Katz, and V. N. Padmanbhan, "The effects of asymmetry on TCP performance," *Mobile Networks and Applications*, vol. 4, pp. 219–241, 1999. 105

[94] H.-W. Lee and S. Chong, "Downlink Resource Allocation in Multi-Carrier Systems: Frequency-Selective vs. Equal Power Allocation," *Proc. IEEE WoWMoM*, Helsinki, Finland, Jun. 2007. 111

[95] "Guidelines for Evaluation of Radio Transmission Technologies for IMT-2000," *Rec. ITU-R M.1225*, 1997. 117

[96] T. Mahmoodi, V. Friderikos, and H. Aghvami, "Allowing Short-Lived TCP Sessions to Ramp-UP in Broadband Wireless Networks," *BWA workshop, GLOBECOM'09*, Hawaii, USA, Dec. 2009. 132

[97] R. J. La and V. Anantharam, "Utility-Based Rate Control in the Internet for Elastic Traffic," *IEEE/ACM Trans. Net.*, vol. 10, pp. 272–286, Apr. 2002. 133

[98] T. Mahmoodi, V. Friderikos, and H. AGhvami, "TCP-aware Resource Allocation in Wireless Networks via Utility Maximisation," *WWRF 22th*, Paris, France, May 2009. 134

[99] "OPNET Technologies," *http://www.OPNET.com*. 151

[100] "The Network Simulator," *http://www.isi.edu/nsnam/ns/*. 151

[101] "GloMoSim," *http://pcl.cs.ucla.edu/projects/glomosim/*. 151

Appendix A

Details and Proofs of the used Mathematical Models

A.1 Newton's Method in Solving Optimisation Problems

Newton's method, named after Isaac Newton and Joseph Raphson, is the well-known method for finding successively better approximations to the roots of a real-valued function. It can be used to solve the unconstrained optimisation problem

$$\text{minimise } f(x)$$

where $f : R^n \rightarrow R$ is convex and twice continuously differentiable. Given the assumption that the problem is solvable, the necessary and sufficient condition for a point x^* to be optimal is, $\nabla f(x^*) = 0$. Newton's method is known to converge quickly, if the initial point of iterations is not far from the desired root x^*.

A.1.1 The Newton step

For $x \in \mathbf{dom} f$, the vector $\Delta x_{nt} = -\nabla^2 f(x)^{-1} \nabla f(x)$ is called the Newton step. Positive definitions of $\nabla^2 f(x)$ implies that,

$$\nabla f(x)^T \Delta x_{nt} = -\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x) < 0$$

unless $\nabla f(x) = 0$. Thus, the Newton step has a descent direction unless x is optimal. The Newton step is motivated in several ways ([75], Chapter 9.5).

Firstly, the second-order Taylor approximation $\widehat{f}$ of f at x is,

$$\widehat{f}(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v, \tag{A.1}$$

which is a convex quadratic function of v, and is minimised when $v = \Delta x_{nt}$. Hence, Newton step is what should be added to x in order to minimise the second-order approximation of f at x. It has been shown that when the function f is quadratic, $x + \Delta x_{nt}$ is the exact minimiser of f, and if f is nearly quadratic, $x + \Delta x_{nt}$ is a very good estimate of the minimiser of f.

The Newton step is also the steepest descent direction at x ([75], Chapter 9.4), for the quadratic norm defined by the Hessain $\nabla^2 f(x)$,

$$\|u\| = (u^T \nabla^2 f(x) u)^{1/2}$$

Finally, the Newton step is what must be added to x so that the linearised optimality condition holds. Given x^* the desired root, if the optimality condition $\nabla f(x^*) = 0$ linearised near x,

$$\nabla f(x+v) \nabla f(x) + \nabla^2 f(x) v = 0$$

which is a linear equation in v, with solution $v = \Delta x_{nt}$.

A.1.2 Newton's method

Algorithm 7 Newton's Method iterations

Given a start point $x \in f$, tolerance $\varepsilon > 0$
repeat

1. Compute the Newton step and decrement.
   $\Delta x_{nt} = \nabla^2 f(x)^{-1} \nabla f(x)$,
2. Stopping criterion: quit if $\|\nabla f(x)\| \leq \varepsilon$.
3. Update: $x = x + \Delta x_{nt}$

A.2 Concept of the Logarithmic Barrier Function

The goal of Logarithmic barrier method is to approximately formulate the inequality constrained problem as an equality constrained problem or unconstrained problem to which e.g., Newton's method can be applied. Given the following optimisation problem,

$$\begin{aligned} &\text{minimise } f_0(x) \\ &\text{subject to: } f_i(x) \leq 0 \quad \forall i \in \{1, ..., m\} \\ &Ax = b \end{aligned} \tag{A.2}$$

first step is to reformulate problem (A.2) to implicit its inequality constraints in the objective function ([75], Chapter 11.2).

$$\begin{aligned} &\text{minimise } f_0(x) + \textstyle\sum_{i=1}^{m} I_-\left(f_i(x)\right) \\ &\text{subject to: } Ax = b \end{aligned} \tag{A.3}$$

where $I_- : R \to R$ is the indicator function for non-positive real,

$$I_-(u) = \begin{cases} 0 & u \leq 0 \\ \infty & u > 0. \end{cases}$$

Although problem (A.3) has no inequality constraint, but its objective is not differentiable, thus Newton's method can not be applied. To this end, the key idea of the barrier method is to approximate the indicator I_- by a differentiable function,

$$\widehat{I}_- = -1/t \log(-u), \;\; \textbf{dom}\, \widehat{I}_- = -R_{++}$$

where $t > 0$ is a parameter that sets the accuracy of the approximation. Like I_-, $\widehat{I}_-$ is convex and nondecreasing, and takes on the value of ∞ for $u > 0$. Unlike I_-, however, $\widehat{I}_-$ is differentiable and closed, i.e., it increases to ∞ as u increases to 0. Substituting $\widehat{I}_-$ for I_- in the objective function (A.3) gives the following approximation,

$$\begin{array}{ll} \text{minimise} & f_0(x) + \sum_{i=1}^{m} -(1/t)\log(-f_i(x)) \\ \text{subject to:} & Ax = b \end{array} \tag{A.4}$$

The objective of (A.4) is convex, since $-(1/t)\log(-u)$ is convex and increasing in u, and differentiable. The function

$$\Phi = \sum_{i=1}^{m} \log\left(f_i(x)\right), \tag{A.5}$$

with $\Phi = \{x \in \textbf{dom}\, R^n | f_i(x) < 0, i = 1, ...m\}$, is called the *logarithmic barrier* for problem A.2. Clearly, problem (A.4) is an approximation for problem (A.3), thus the question arises that how well it can approximate the solution of (A.3). It can be shown that, as t increases, the approximation becomes more accurate. On the other hand when t is large, the Hessian of function $f_0 + \frac{1}{t}\Phi$ varies rapidly near the boundary of the feasible set. Therefore, it is difficult for the Newton's method to minimise the objective.

A.2.1 Example Problem

$$
\begin{aligned}
\min \quad & f(X) = 6(X_1-10)^2 + 4(X_2-12.5)^2 & \\
\text{s.t.} \quad & g1(X) = {X_1}^2 + (X_2-5)^2 & \leq 50 \\
& g2(X) = {X_1}^2 + 3{X_2}^2 & \leq 200 \\
& g3(X) = (X_1-6)^2 + {X_2}^2 & \leq 37
\end{aligned}
$$

Adding the constraints to the objective function using the logarithmic barrier function, reformulates the problem as,

$$
\begin{aligned}
\min \quad & 6(X_1-10)^2 + 4(X_2-12.5)^2 \\
& -1/t\,(\log(-g1(X)+50) + \log(-g2(X)+200) + \log(-g3(X)+37))
\end{aligned}
$$

This can be solved using the Newton method with one feasible initial point. Given the new objective function $F(x) = f(X) - \frac{1}{t}\Phi(X)$, the Newton iterations are as follows,

1. choose x_0
2. $x_k{=}x_0$
3. compute $\nabla F(x_k)$ and $\nabla^2 F(x_k)$
4. $x_{k+1} = x_k - \nabla^2 F(x_k)^{-1} \cdot \nabla F(x_k)$
5. If $\|\nabla F(x_{k+1})\| \leq \varepsilon$ stop
   else go back to 2.

As an initial point, a feasible point $x_0 = [6, 5]$ is chosen, thus the first and the second gradient of the objective function are computed and the optimal point is found in the iterative loop.

Appendix B

Simulation Modelling

B.1 OPNET Modeler

B.1.1 Why OPNET Modeler

OPNET modeler is, at its core, a Discrete Event Simulation (DES) environment. Packets and protocol dynamics are all explicitly modelled and scheduled as sequential events at specific simulation time instances, which are managed by the simulation kernel [99]. A protocol modelled in a DES simulation, behaves the same way as in real-life production environments, as any protocol can be modelled in the same intricate detail. While other alternatives, e.g. flow analysis, are generally faster in execution, and can be used to study the functionality of some part of the wireless system, they will not reflect the functionality of the whole protocol stack TCP timer expiration on wireless data transfer. Moreover, as the main focus of this thesis is on the performance of TCP, the real implementation of TCP with its detail functionalities such as timer expiry, were required.

Among several available DES applications, such as NS-2 [100], and GloMoSim [101], OPNET modeler [99] is selected for the investigations of this thesis because of the followings. The OPNET simulation platform provides very solid, robust and scalable simulation kernel, it has integrated graphical environments, including network editors and result analysis tools, and also huge standard model library, ranging from detailed wireless communication modelling to the various transport layers including many flavours of TCP. Moreover, most of the standard library models are open source, and well documented to allow modifications. The last, and the most important is that OPNET modeler is an industry-standard commercial application at affordable university licensing with available technical support, which makes it a reliable and useful application.

B.1.2 Modelling Methodology

OPNET has three distinct levels for modelling the communication systems, and each level has an associated editor to work in. The first one is Network Editor, in which the network topology can be built and individual nodes and connections can be configured. The second one is Node Editor, in which one can edit each node of the network and depict the protocol stack of the node based on its standard. The third and also the lowest level is Process Editor, in which individual protocols of the network are modelled. An example showing all three levels is plotted in Figure B.1.

B.1.2.1 Network Editor

In Network Editor, the highest level overview of the communications network is provided. It is where the objects are placed, configured, given a physical location, and possibly even interconnections with other objects. Sub-networks, nodes and links are the main building blocks of any network. Nodes are the main building blocks of any network model. They are categorized into terminal nodes and communication nodes, and can be any type of device in the networking context. Therefore they are not necessarily sending or receiving traffic. Links form the connections between the wired nodes in the network, and have a wide variety of types.

B.1.2.2 Node Editor

Nodes are the fundamental building blocks of any network. They can perform a wide range of functions including workstations, routers, Layer 2 hubs. Using Node Editor, the internal structure of any node is constructed by placing different protocol modules inside the node and connecting them. This intuitive block-structured approach is similar to how we think of any protocol stack: the function of each layer (module) is well-defined and it often only needs to communicate with the neighbor layers. Communication nodes usually deal with packets. Therefore the modules inside the node will receive packets, process them, and send them out to another module or discard them. The WLAN work station advance node from the OPNET Standard Library is shown in Figure B.1. It can be noticed that each module (except CPU) is connected to another module through so-called Packet Streams. The standard TPC/IP stack from the lowest physical layer to the application layer can be easily recognized. Although the most intuitive way to transfer data packets between the modules is using Packet Streams, this is not the only way. Normally each module can deliver any type of packet to any other module that is not even inside the same node without using Packet Streams, which is a less transparent method.

Another way of connecting the modules to each other is through Statistic Wires. A set of statistics such as packet arrival rate, buffer usage, or SNR can be dedicated to any module. Therefore, a module can be interrupted whenever a specific statistic is updated at the connected module. This update can be either variations in the value of the parameter, or crossing a certain threshold. For example, this method is extensively used in the WLAN model; MAC layer uses Statistic Wires to monitor if the channel is busy, or to get information on the received packet's power level. Finally, the two wireless modules connected to the MAC layer, are the physical interface of the node. While all other modules can be fully defined by the user, these two transceivers are tied to a more rigid structure. The wireless pipeline stages that are described later, control the functionality of these transmitter and receiver ports. In order to model custom antenna patterns, wireless transceivers can be connected to an antenna module. When the antenna is not specified, however, the default isotropic model with zero dB gain in all directions is assumed.

B.1.2.3 Process Editor

Processes are used for specifying the functionality and behavior of various modules in the protocol by the use of Finite State Machine (FSM)s. The procedures in each state are programmed using the C language, and also OPNET developed functions called Application Programming Interface (API). Note that C++ can be used as well, but all of OPNET's code is in C style. Every state of the FSM has two different code blocks: Enter Execs block and Exit Execs block. The former is executed whenever a state transition takes control to this state and the latter is executed whenever a state transition out of this state (including self-transitions) occurs. At each state transition three code blocks executed: Exit Execs of the originating state, the transition code executive and Enter Execs of the arriving node. It must be noted that any of these code blocks can be empty.

A state is either Forced (marked as green) or Unforced (marked as red). Forced states transit to the new state after their execution without waiting for any interrupt. Unforced states do wait for any form of interrupt, either from inside this module, another module or even another node. Therefore, FSMs are interrupt-driven due to their construction: a process will indefinitely remain in its most recent Unforced state if no interrupt with valid transition is received. Possible interrupts consist of packet receptions (either through Packet Streams or directly delivered from another process), statistic interrupts (through Statistic Wires), or any form of interrupt, locally or remotely scheduled. As most events only contain an identification number, it would be useful to associate some more contextual information with the events. Interface Control Information (ICI)s are structures that can be defined by user, and used to convey any form of information such as simple type values, and memory pointers to the receiving end of the scheduled interrupt.

B.1.3 Wireless Modelling

Each transmitted packet passes through a series of blocks that model the effects of so called wireless channel. Each block is responsible for one specific propagation effect like delay, noise, error, which are called pipeline stages. These pipelines along with the modifications that provided for the purpose of this thesis are further discussed here.

B.1.3.1 Wireless Channels

The effect of wireless channel is modelled via 12 pipeline stages as follows. These pipelines and their order is depicted in Figure B.2.

1. **Transmission delay** is the time needed for the entire packet to complete transmission onto the medium.

2. **Link closure** is the ability for the packet to reach the receiver channel. The use of this stage is eliminating the receivers that are totally unable to receive anything in that specific transmission (due to jamming, obstacles, etc.).

3. **Channel match** is limitation of correct reception of data packets only to the channels that match the channel characteristics of the transmitter. Although some other channels are not able to receive the data contained in the packets, it is still possible that their performance be influenced by this transmission. For example if their frequency bands partially overlap, they may consider this packet as noise.

4. **Tx antenna gain** is taken into account for calculation of the transmission power, based on the antenna pattern of the transmitting node.

5. **Propagation delay** is the amount of time needed for a packet to traverse the medium.

6. **Rx antenna gain** is taken into account when calculating the received power.

7. **Received power** determines the received power of the transmitted signal. Effective parameters in the calculation are transmitted power, antenna gains, and wireless channel losses such as path loss and fading effects.

8. **Background noise** models of any form of in-band noise including thermal or galactic noise, and neighboring electronics.

9. **Interference noise** is a model for the effect of concurrent receptions at the same channel. It includes packets marked as noise in the channel match stage. It must be noted that any packet can be interrupted any number of times when other packets arrive at the receiver, also that different segments of a packet can have different noise figures.

10. **Signal to Noise Ratio** (SNR) is the combination of the effects of the three previous stages. It determines the total average SNR for a packet.

11. **Bit Error Rate** (BER) is the probability of bit errors during each interval of constant SNR. It is calculated based on the SNR value and as a function of the modulation used by the transmitter.

12. **Error allocation** estimates the number of bits in error for each received packet based on the probability produced by the previous stage.

13. **Error correction** is executed after the completion of packet reception and confirmation of its validity, to check if the receiver can recover from the errors encountered by the packet. If positive, the packet will be sent to the higher layer over a Packet Stream. If not, it will be dropped by the simulation kernel.

B.1.3.2 Modifications to the Standard Library

For the simulation investigations of this thesis, two of the pipeline stages are modified: received power, bit error rate. In the received power pipeline, the ITU indoor path loss model (used in Chapter 3), rayleigh fading and Lognormal shadowing effects (used in Chapter 4) are implemented. In the bit error rate pipeline stage, the dynamic selection of FEC is added. The selection of BER for each data transmission is from a look-up table depends on the SNR of the corresponding channel and the FEC rate that is applicable to that data transmission.

B.1.4 Transport Layer Modelling

The TCP model in OPNET is based on its RFCs, thus following features are supported. The process model of TCP can also be seen in Figure B.1.

1. Connection Setup and Termination: Three-way handshake protocol used to establish connections and four-way exchange to close connections.

2. Reliability: Acknowledgments and retransmissions are triggered by adaptively calculated retransmission timers.

3. Flow Control: Dynamic windowing of transmissions based on the availability of buffering resources at the receiving nodes and the middle routers.

4. Detailed implementation of different TCP flavours: Tahoe, Reno, and New Reno with SACK and Window Scaling extensions.

5. Configurable parameters such as MSS, timer values, acknowledgment schemes and extensions options such as, fast-retransmit, fast-recovery, window scaling, and/or SACK.

B.1.4.1 Modifications to the Standard Library: TCP Westwood

Some modifications are applied to the TCP congestion control algorithm so that TCP Westwood is also supported. The implementation of TCP Westwood consists of two main parts: estimation the bandwidth by monitoring the ACK reception rate, updating the cwnd and ssthresh in the event of loss accordingly. After adding these two functionalities to the TCP congestion control, new attribute is added to the TCP process model so that TCP Westwood can be selected as one of the available flavour of TCP from the Network Editor environment.

B.2 Network Configuration

The network configuration that is used in all the simulated scenarios of this thesis, is a single cell scenario where all the mobile users are competing over the same wireless network. The studied network is IEEE 802.11a, for two main reasons, firstly the well-implemented model of WLAN in the OPNET, and secondly using the OFDM technique that the properties associated with, has led to its consideration as a candidate for the new generation of wireless networks.

To be able to specify the characteristics of each end-to-end path, and its corresponding TCP flow independently from the others, in all the simulated scenarios, each mobile users is connected to a unique end-host or server. This network configuration can be seen in Figure B.1, Network Editor.

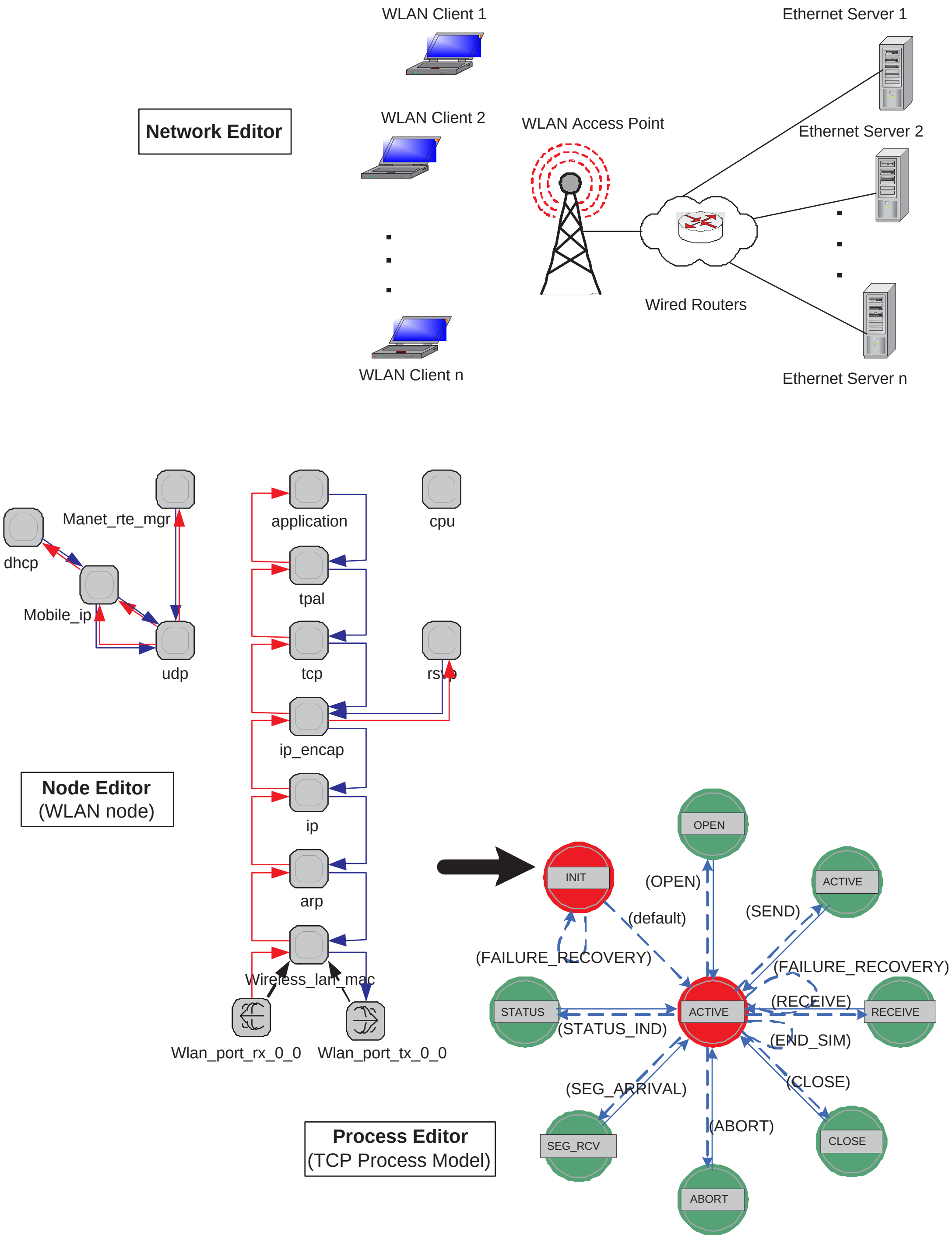


Figure B.1: Hierarchy of the OPNET modelling

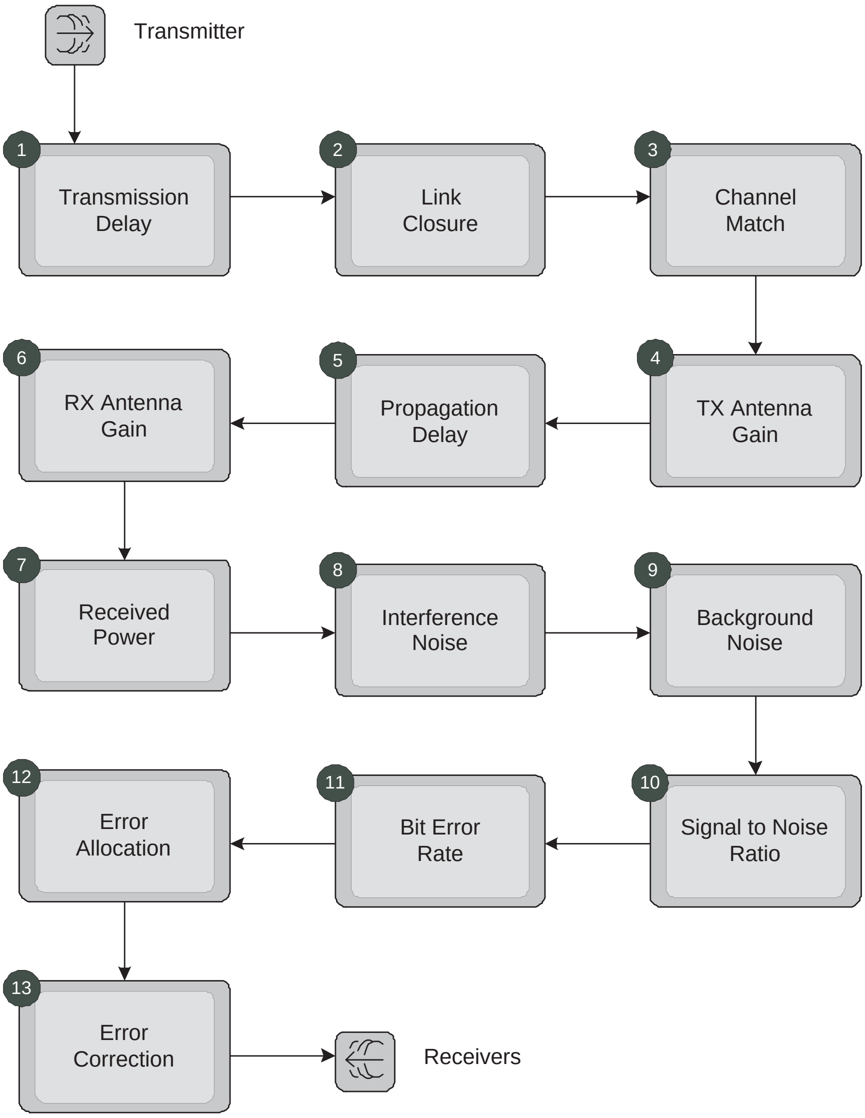


Figure B.2: Wireless Link Pipeline Stages