

**PATH DIVERSIFIED RETRANSMISSION FOR TCP OVER  
MULTI-CHANNEL WIRELESS MESH NETWORKS**

by

Xiaoyuan Guo

B.Eng, Jilin University, 2007

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF APPLIED SCIENCE  
in the School  
of  
Computing Science

© Xiaoyuan Guo 2011

SIMON FRASER UNIVERSITY

Summer 2011

All rights reserved. However, in accordance with the Copyright Act of Canada, this work may be reproduced without authorization under the conditions for Fair Dealing. Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.

## APPROVAL

**Name:** Xiaoyuan Guo  
**Degree:** Master of Applied Science  
**Title of Thesis:** Path Diversified Retransmission for TCP over Multi-Channel Wireless Mesh Networks

**Examining Committee:** Dr. Qianping Gu  
Chair

---

Dr. Jiangchuan Liu, Senior Supervisor

---

Dr. Mohamed Hefeeda, Supervisor

---

Dr. Jie Liang, SFU Examiner

**Date Approved:** May 31, 2011



SIMON FRASER UNIVERSITY  
LIBRARY

## Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the "Institutional Repository" link of the SFU Library website <[www.lib.sfu.ca](http://www.lib.sfu.ca)> at: <<http://ir.lib.sfu.ca/handle/1892/112>>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library  
Burnaby, BC, Canada

# Abstract

Path diversity exploits multiple routes simultaneously, achieving higher aggregated bandwidth and potentially decreasing delay and packet loss. Unfortunately, for TCP, naive load splitting often results in inaccurate estimation of round trip time (RTT) and packet reordering. As a result, it can suffer from significant instability or even throughput reduction. This is particularly severe in multi-channel Wireless Mesh Networks (WMNs), as validated by our analysis and simulation.

To make multi-path TCP viable over multi-channel WMNs, we propose a novel cross-layer design with a smart traffic split scheme, namely, Path Diversified Retransmission (PDR). PDR differentiates the original data packets and the retransmitted packets, and works with a novel Joint Optimization of Bandwidth and Delay multi-path routing protocol, JOBD, to distribute them separately. PDR does not suffer from the RTT underestimation and extra packet reordering, which ensures stable throughput improvement over single path routing. Through extensive simulations, we further demonstrate that, as compared to state-of-the-art multi-path protocols, our PDR with JOBD noticeably enhances the TCP throughput and reduces bandwidth fluctuation, with no obvious impact to fairness.

# Acknowledgments

First and foremost, I thank my supervisor, Dr. Jiangchuan Liu. His significant enlightenment, guidance, encouragement and patience on my research have led to the success of my M.Sc. study. His excellent insights and suggestions have helped me to hurdle all the obstacles in completing this work. I have been extremely lucky to have such a supervisor who cared so much about my work, responded to my questions and queries so promptly and concerned about my life.

I thank my supervisor Dr. Mohamed Hefeeda and my thesis examiner Dr. Jie Liang, for reviewing this thesis carefully and offering useful advice that helped me to make the thesis much better. I thank Dr. Qianping Gu for taking the time to chair my thesis defense. I would also like to express my gratitude to all the faculty and staff in the School of Computing Science at Simon Fraser University.

I thank my colleagues and friends at Simon Fraser University for their kind help and support. I especially thank my colleague Feng Wang, for our happy cooperation on the JOBD paper. I thank my ex-colleague Fan Yang, Fajun Chen and Kunfeng Lai for their great assistance and encouragement on my research. I also thank Yan Ding, Ouldooz Baghban Karimi, Xu Cheng, Haiyang Wang, Haitao Li, Xiaoqiang Ma, Azin Dastpak, Ziwen Zhang and Fei Chen for their help on this thesis. My friends not only offered help in my studies but also in my daily life. I deeply appreciate these.

Last but not least, I thank my family for their selfless love, concern and support: my parents, my uncles, my aunts, my brothers and sisters. I could not achieve this success without you. This thesis is dedicated to you all. I love you all!

*To my family and friend*

*For poise, walk with the knowledge that you never walk alone*

# Contents

<b>Approval</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>iv</b>
<b>Dedication</b>	<b>v</b>
<b>Quotation</b>	<b>vi</b>
<b>Contents</b>	<b>vii</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.1.1 Wireless Mesh Networks . . . . .	1
1.1.2 Multiple path TCP . . . . .	2
1.2 Motivation . . . . .	3
1.3 Contributions . . . . .	4
1.4 Thesis Organization . . . . .	5
<b>2 Related Work</b>	<b>6</b>
2.1 TCP over Multiple Paths . . . . .	6
2.2 Path Diversity in Wireless Mesh Networks . . . . .	7



2.3	Multi-QoS Routing . . . . .	8
<b>3</b>	<b>Understand TCP Performance with Multipath</b>	<b>9</b>
3.1	Simulation Environment . . . . .	9
3.2	Simulation Results and Analysis . . . . .	10
3.3	Summary . . . . .	12
<b>4</b>	<b>Path Diversified Retransmission (PDR) and System Design</b>	<b>13</b>
4.1	PDR . . . . .	13
4.2	Improvement of TCP Throughput with PDR . . . . .	15
4.3	System Design . . . . .	17
4.4	Summary . . . . .	19
<b>5</b>	<b>Joint Optimization of Bandwidth and Delay (JOBDD) Routing Protocol</b>	<b>20</b>
5.1	System Overview and Problem Statement . . . . .	20
5.2	Bandwidth and Delay Estimation under Channel Diversity . . . . .	24
5.2.1	Available Bandwidth under Inter-Flow Interference . . . . .	24
5.2.2	Available Bandwidth under Intra-Flow Interference . . . . .	26
5.2.3	Available Bandwidth over a Multi-Channel Path . . . . .	27
5.2.4	End-to-End Delay over a Multi-Channel Path . . . . .	27
5.3	JOBDD Routing Protocol Design . . . . .	30
5.3.1	Multi-Channel AOMDV . . . . .	30
5.3.2	Route Discovery in JOBDD . . . . .	31
5.3.3	Parameter Estimation . . . . .	32
5.4	Summary . . . . .	33
<b>6</b>	<b>Performance Evaluation</b>	<b>34</b>
6.1	Simulation Environment . . . . .	34
6.2	Simulation Results . . . . .	35
6.2.1	Probability of Path Overlapping . . . . .	35
6.2.2	Throughput Results . . . . .	35
6.2.3	Fairness Results . . . . .	38
6.3	Summary . . . . .	39

<b>7 Conclusion and Future Work</b>	<b>40</b>
7.1 Conclusion . . . . .	40
7.2 Future Work . . . . .	40
<b>Bibliography</b>	<b>41</b>

# List of Tables

5.1 List of Notations . . . . .	23
---------------------------------	----

# List of Figures

3.1	Comparisons of TCP throughput . . . . .	10
3.2	Packet reordering in multiple path routing . . . . .	11
4.1	A network with 8 nodes. The available bandwidth and the end-to-end delay of each path are shown along each path. . . . .	13
4.2	TCP loss recovery process in single path routing . . . . .	14
4.3	$d_1$ and $d_2$ are the loss recovery times of TCP over a single path and that with PDR, respectively. If the retransmitted packets are sent on the path with minimum delay, we have $d_2 < d_1$ . . . . .	15
4.4	Evolution of window size over time when retransmission is triggered by three-duplicate ACKs . . . . .	16
4.5	Overview of the cross-layer implementation . . . . .	18
5.1	A multi-channel WMN with 3 nodes. Each node is configured with two channels, and thus there are two links between neighboring nodes. . . . .	21
5.2	A multi-channel WMN with 3 nodes, $n_1$ , $n_2$ and $n_3$ . Each node is configured with two interfaces. Each interface is considered as an I-node and connected to its M-node by an edge with infinite bandwidth and zero delay. Bandwidth and delay on each edge are shown. We assume $bw_1 > bw_2$ , $d_1 > d_2$ , $bw_3 > bw_4$ , $d_3 > d_4$ . If $n_1$ is the source node and $n_3$ is the destination, the maximum-bandwidth path will be $n_1 \rightarrow n_1^1 \rightarrow n_2^1 \rightarrow n_2 \rightarrow n_2^2 \rightarrow n_3^2 \rightarrow n_3$ and the minimum-delay path will be $n_1 \rightarrow n_1^2 \rightarrow n_2^2 \rightarrow n_2 \rightarrow n_2^1 \rightarrow n_3^1 \rightarrow n_3$ . . . . .	22

5.3	Example of inter-flow interference between two paths. Assume links $m_2$ , $m_5$ and $n_1$ are assigned with the same channel. The small circle covers available transmission range of a node, and the large circle denotes interference range of a node. So links $m_2$ and $n_1$ will suffer from co-channel inter-flow interference, while $m_5$ and $n_1$ have no such interference. . . . .	24
5.4	JOB D RREQ packet header . . . . .	31
5.5	The route list of AOMDV and JOB D . . . . .	32
6.1	Probability that the bandwidth- and delay- optimized paths are overlapping . . . .	36
6.2	Comparisons of TCP throughput . . . . .	37
6.3	Comparisons of TCP throughput fluctuation . . . . .	38
6.4	Comparisons of TCP fairness index . . . . .	39

# Chapter 1

## Introduction

In this chapter, we first introduce the background of wireless mesh network and TCP performance in multipath routing. Then we present our motivation and summarize our contributions in this thesis. Finally we describe the organization of this thesis.

### 1.1 Background

#### 1.1.1 Wireless Mesh Networks

Wireless mesh network(WMN) is an evolved multihop wireless network, in which radio nodes are dynamically self-organized and reconfigured in a full or partial mesh topology. The nodes in WMNs can be classified into two categories: mesh routers and mesh clients. Mesh routers are usually equipped with multiple interfaces built on the same or different wireless access technologies, which are different from conventional wireless routers. They have minimal mobility, interconnect to each other and form a relatively static mesh backbone for mesh clients. Mesh clients, such as laptops and PDAs, may have more mobility. They could also work as routers. However, their hardware platform and software can be much simpler than that for mesh routers. For instance, a mesh client may just need a single wireless interface and do not expect gateway/bridge functions.

Apart from providing network access for mesh clients, mesh routers with their gateway/bridge functionalities enable WMNs to be integrated with heterogeneous networks, *e.g.*, Internet, Wi-Fi networks, Cellular networks. For conventional nodes with wireless interface cards, they can connect to WMNs directly through mesh routers. For those wired clients, they could access WMNs by other networks, for example, Ethernet. Hence, with WMNs people can access network any time, any

where.

WMNs provide static infrastructure networks, which brings many advantages in terms of low up-front cost, quick network deployment, easy network maintenance, reliability, and robustness. Moreover, the technology of multiple channels are employed in WMNs to further improve network capacity. As a result, in addition to being widely acceptable in the traditional application scenarios of ad hoc networks, WMNs are undergoing widespread commercial developments for both military and civilian uses. Currently, U.S. military use WMNs to connect their computers. They are also used for electric meters monitoring and in the one laptop per child program. Moreover, WMNs can be used to deploy broadband home networking, building automation, highspeed metropolitan area networks, etc. In particular, WMNs are used in the old buildings without wired networks in big cities, as it is difficult to deploy wirelines in those buildings while WMNs can be laid out much more easily and costlessly.

### 1.1.2 Multiple path TCP

Path diversity exploits the available paths in a network, and maintains multiple routes for each flow simultaneously. It is predominant in wireless networks instead of single path routing, profiting from its ability to aggregate bandwidth, enhance routing robustness, and decrease delay and packet loss. The insight reason is that path diversity splits traffic into several paths, achieving load balance by unitizing available network resource efficiently.

The design of TCP on its error detection and congestion control mechanism is based on the premise that packet loss is caused by network congestion. Therefore, TCP sender shrinks its congestion window when loss is detected. TCP detects loss with two strategies, retransmission timeout (RTO) and TCP sequence number. The sender sets up a retransmission timer and consecutive sequence number for each segment sent out. When the retransmission timer is expired in the sender side or out-of-order packet is received in the receiver and duplicate ACKs are sent back, TCP sender considers it as an indication of network congestion, and therefore adopts *slow-start* or *fast retransmit* algorithm to adjust its congestion window.

Given the potentially increased bandwidth, path diversity would improve TCP performance. However, with a naive traffic split, TCP could suffer from more severe throughput degradation and fluctuation when runs over multiple paths. Specifically, TCP was designed to deliver packets in order and avoid timeout within certain time interval along a single path, which is not guaranteed

in multipath routing since different paths have diverse delay/bandwidth characteristics. The problem is further aggravated with the out-of-order packet delivery and persistent timeout events across multi-path, which TCP treats as a sign of network congestion imitatively. In this case, TCP would spuriously retransmit packets, making its congestion window unnecessarily small, and thus resulting in significant throughput reduction and fluctuation [41].

## 1.2 Motivation

As infrastructure multiradio multihop wireless networks, wireless mesh networks (WMNs) are easy to deploy and maintain, potentially developed in spreading Internet accesses and other network services in personal, local, and old metropolitan areas. Unfortunately, the end-to-end performance is very poor in WMNs due to the limited bandwidth, strong interference among the wireless nodes because of broadcast feature of wireless medium, and unavoidable collision caused by hidden and exposed terminals, particularly with multi-hop routing.

Multiple path transmission could obtain higher aggregated bandwidth and potentially decreases delay and packet loss. WMN is a typical network offering several paths from a source to a destination and a series of multipath protocols have been proposed for WMNs routing [6, 59, 69]. Yet the interaction and the best combination between the transport protocols and the multipath routing protocols are still unclear, particularly for TCP. We are interested in TCP as it is expected to be the dominant transport protocol in WMNs for its simplicity, reliability, and built-in adaptiveness and fairness. Moreover, the existence of Network Address Translators (NATs) and firewalls further makes TCP the preferred choice over UDP for many ISPs and network administrators.

Unfortunately, TCP could suffer from severe performance degradation in multiple path routing mainly due to various path delay in different routes. Several proposals were suggested to improve multipath TCP performance [36, 55, 58]. They either modify the conventional TCP congestion control mechanism to be immune to packet reordering [58], or split traffic carefully [36, 55]. However, they are all relatively complex and mainly focus on how to deal with packet reordering with few considerations of extra timeout events in multipath TCP. To address the above challenges, we intend to exploit deeply into the problem, and design an efficient transmission scheme to make multipath TCP viable over WMNs.



### 1.3 Contributions

Our contributions are summarized as follows.

- we propose a novel cross-layer design with a smart traffic split scheme, namely, Path Diversified Retransmission (PDR). PDR differentiates the original data packets and the retransmitted packets, taking into account the information from both the transport layer and the network layer. The key observation is that the amount of retransmitted packets is relatively smaller, but they have more stringent delay requirement. Therefore, the original packets and the retransmitted packets should follow different paths with different optimization criteria, i.e., bandwidth maximization and delay minimization. We demonstrate that this design achieves higher throughput and lower delay, and more importantly, it largely eliminates packet re-ordering and extra timeouts.
- To cooperate with PDR, We develop JOB (Joint Optimization of Bandwidth and Delay) protocol, which extends the conventional Ad hoc On-demand Multipath Distance Vector (AOMDV) routing protocol [47] (In fact, PDR can be integrated with any QoS multipath routing protocols in WMNs). JOB offers the maximum-bandwidth path and the minimum-delay path simultaneously.
- During the development of JOB, we show that, as long as the best channels with different QoS metrics are not overlapped between WMN node pairs, there exist complete disjoint paths with different QoS metrics. We present efficient solutions to discover such paths, particularly for bandwidth- and delay-optimization. Note that existence of multiple channels notably changes the interference pattern from that in single-channel WMNs, rendering the existing bandwidth and delay estimate tools [22, 26, 43, 83] far from being accurate. Our JOB (Joint Optimization of Bandwidth and Delay) protocol offers novel estimation tools that take both intra- and inter-flow interference under channel diversity into account, achieving much more accurate bandwidth and delay estimation in the multi-channel WMN environment.

Our work have been published in the *Multimedia Systems Journal* in Nov. 2010 [80] and *IEEE International Workshop on Quality of Service* in June 2010 [29].

## 1.4 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 presents an overview of related work systematically. In Chapter 3, we verify the reduction of TCP performance over multipath routing via *ns-2* simulation and exploit the insightful reasons deeply. Chapter 4 proposes PDR and analyzes its potential improvement for TCP performance as well as our system design. Chapter 5 presents the design and implementation of JOBD in detail, in which we analyze the joint optimization problem of bandwidth and delay in multi-channel multi-path WMNs, develop new algorithms to estimate bandwidth and delay under channel diversity and discuss the practical implementation issues of JOBD protocol. Chapter 6 evaluates our proposed protocol with *ns-2*. Finally, Section 7 concludes the thesis and provides some future directions.

## Chapter 2

# Related Work

In this chapter, we provide an overview of the related work systematically. In general, we present the related work in the following three aspects: TCP over multiple paths, path diversity in WMNs as well as multiple QoS routing.

### 2.1 TCP over Multiple Paths

TCP suffers from performance degradation in WMNs as in other wireless networks, due to its inability to differentiate congestion losses and non-congestion losses with link errors [5]. Plenty of TCP variants were suggested to cope with this problem [15, 24, 82]. Cross-layer design typically adjusts TCP congestion window according to the information measured from the MAC layer to identify congestion losses [18, 34]. These schemes all keep a single path to each destination without utilizing available paths effectively.

The Multipath TCP (MPTCP) working group aims to support regular TCP over multiple paths without significant modifications to the existing Internet infrastructure. Current drafts include proposals to extend the traditional TCP for multipath operation with multiple addresses [23, 31], to specify a new address family in the socket API for the multipath interface [61], and to provide interfaces accessing to multipath information for applications [63]. Most of them remain in draft status, and have not addressed TCP in WMNs.

Unfortunately, it is known that, with multipath routing, TCP could suffer throughput fluctuation or even reduction due to inaccurate RTT estimation and packet reordering [36, 55, 58]. Proactive reordering algorithms were proposed in [55] and [58] to mitigate out-of-order delivery. Kandula et

al. [36] proposed Flare to divide and cache traffic at the granularity of packet bursts, with the observation that if the time between two successive packets is larger than the maximum delay difference between multiple paths, the subsequent packets can be scheduled on any path with no risk of reordering. SPRIC [78] sorts reordered packets in the same block with interrupt coalescing to eliminate or reduce packet reordering at the destination. Other solutions include dynamically adjusting the TCP reordering threshold to avoid improper retransfer or quickly recovering from spurious congestion window shrinks [14, 86]. All these solutions are relatively complex, while our PDR is simpler with built-in mechanism to avoid re-ordering. Our solution further explores cross-layer design with QoS optimization for TCP-based applications [17, 48, 89].

## 2.2 Path Diversity in Wireless Mesh Networks

Path diversity has long been recognized as an important approach toward delay reduction, load balance, and fault tolerance. One critical concern of path diversity is that the multiple paths may share joint links, which will be a bottleneck. Wang and Crowcroft [75] first proved that multi-constraint-path selection in single-path routing is a NP-complete problem and gave three heuristic path computation algorithms. For multi-path routing, Wang *et. al* [74] argued that the paths should be disjoint, otherwise network performance enhancement would be bottlenecked. Fully disjoint paths (node-disjoint and link-disjoint paths) however are not always available in a network. Various algorithms therefore have been proposed to compute maximum node- and link-disjoint paths [76, 81]. A typical example is AOMDV [47], which extends the Ad hoc On-demand Distance Vector (AODV) single path routing protocol to compute multiple disjoint paths using controlled flooding. Recently, there have also been efforts toward multipath routing in WMNs [6, 59, 69], benefiting from multi-radio, multi-channel, multi-gateway and opportunistic transmission additionally. It is believed that multipath routing would be better to incorporate QoS attributes [55].

Multi-channel has been suggested as an important technique to significantly reduce the wireless interference and improve network capacity in WMNs [4]. Much effort has been devoted to utilizing multi-channel in multi-path routing in this context [6, 59, 68, 69, 85], but mainly focus on homogeneous QoS. New routing metrics that capture the intra/inter-flow interference under channel diversity in WMNs have also been examined in [22, 26, 43, 83]. WCETT (Weighted Cumulative Expected Transmission Time) [22] combines link weights with ETT (Expected Transmission Time) to account for intra-flow interference, where ETT is derived based on link loss rate and bandwidth. WEED (Weighted End-to-End Delay) [43] estimates the end-to-end delay with respect to intra-flow

and inter-flow interference through a sub-path abstraction. To the best of our knowledge, our work first employs multi-channel to utilize bandwidth- and delay-optimized disjoint paths simultaneously in WMNs.

### 2.3 Multi-QoS Routing

QoS routing in the domains of throughput and timeliness were addressed in [46, 48, 89]. These approaches all assume QoS metrics, such as bandwidth, delay, and packet loss rate, are known. In [17], bandwidth is estimated by two methods. One is ‘listen’ and the other is ‘hello’. The first method measures delay by monitoring the channel idle time. The second is based on the information provided in ‘hello’ message. The ‘listen’ method is more suitable for WMNs because of shared transmission medium in wireless networks [30].

There have been significant studies on finding routes with diverse QoS guarantees in both single-path routing [46, 75] and multi-path routing [16, 49, 87]. As said above, in multi-path routing, fully disjoint paths are not always available in networks. Maximally disjoint QoS paths are therefore suggested as an approximation and various heuristic algorithms were proposed [16, 49, 87]. Yang *et. al* [49] raised a problem that finding multiple paths, among which the delay of the longest path is minimized and the aggregated bandwidth of the multiple paths meets the bandwidth requirement, and gave a pseudo-polynomial time algorithm. Soysa *et al.* [87] also brought a heuristic solution for the NP-complete problem that providing multiple connections with predefined reliability requirement and differential delay constraint. Furthermore, the interference among multiple paths, a key factor in wireless transmission, has not been addressed in these works. Instead, we would consider both inter-flow and intra-flow interference in our JOBD routing protocol design.

## Chapter 3

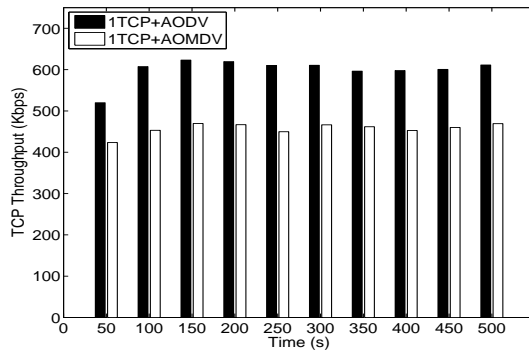
# Understand TCP Performance with Multipath

Before presenting PDR, we first conduct a series of *ns-2* experiments to examine the TCP performance with multipath routing in WMNs and understand the insightful reasons that affect multipath TCP performance enhancement deeply.

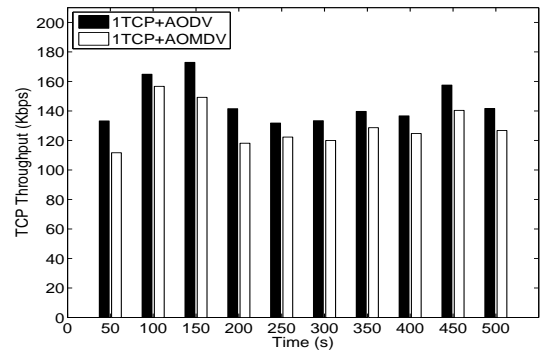
### 3.1 Simulation Environment

We ran TCP-Veno, a well-known wireless TCP version, over both AODV and AOMDV routing protocols. AODV is a widely implemented single path routing protocol for WMNs, while AOMDV transmits data through two paths simultaneously with even split. We adopt a similar configuration as in [43, 57, 79]. Specifically, we established a WMN with 40 nodes placed randomly in a  $1000\text{m} \times 1000\text{m}$  square field. Each node has a radio propagation range of 250m and its carrier sensing distance is 550m. The MAC protocol adopts 802.11a with 5Mbps as its channel rate. We use the channel assignment scheme proposed by P. Kyasanur *et al.* [39] and configure 4 interfaces and 12 channels for each node. Note that AODV and AOMDV both take hopcount as their routing metric and we extend them to run over multi-channel networks (The details in protocol extending would be presented in Chapter 5).

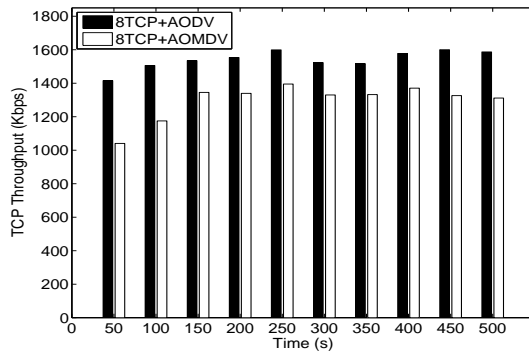
We simulated 3 scenarios with different traffic patterns to examine TCP behavior in various environments, including 1 TCP flow, 1 TCP and 7 UDP flows, and 8 TCP flows. For each scenario, 30 runs with different random seeds were executed, and the results are then averaged to mitigate



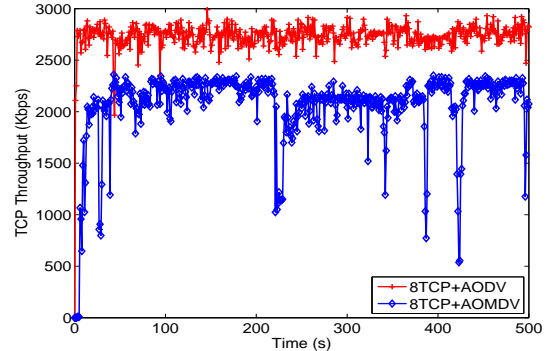
(a) TCP throughput (1 TCP flow). The average TCP throughputs for AODV and AOMDV are 599.6Kbps and 457.2Kbps, respectively.



(b) TCP throughput (1 TCP and 7 UDP flows). The average TCP throughputs for AODV and AOMDV are 147.3Kbps and 125.8Kbps, respectively.



(c) TCP throughput (8 TCP flows). The average TCP throughputs for AODV and AOMDV are 1341.2Kbps and 1296.5Kbps, respectively.



(d) TCP throughput (8 TCP flows in one run)

Figure 3.1: Comparisons of TCP throughput

randomness. The performance is measured in terms of TCP throughput.

## 3.2 Simulation Results and Analysis

Fig. 3.1 shows the average TCP throughput in the three scenarios, respectively. Surprisingly, in all the scenarios, TCP performs poorer with multiple paths than with a single path. This challenges the conventional belief that bandwidth aggregation from multiple path routing is beneficial [6, 59, 69]. A closer look suggests that these existing multipath protocols are mainly designed for UDP, with few explicit optimizations for TCP. As a reliable transport protocol, TCP was designed to deliver packets in order through a single path. To ensure reliable transmission, it uses sequence numbers to

identify the order of bytes sent from the source. The destination expects in-order segments and if it receives a packet with an unexpected sequence number, it buffers the out-of-order packet and returns a duplicate ACK to the source. Retransmission is triggered in the sender side with 3 duplicate ACKs or a timeout. Multiple path routing, however, would introduce extra timeout and packet reordering events because of delay difference in multiple paths, resulting in TCP spuriously retransmitting segments and keeping its congestion window unnecessarily small [41]. Consistent with the previous studies [55, 84], we have the following observations:

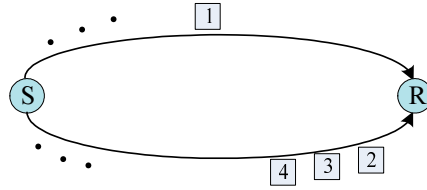


Figure 3.2: Packet reordering in multiple path routing

1. Packets on slow paths may suffer from timeout persistently because of RTT underestimation. For illustration, assume there are  $n$  paths between a source and a destination,  $r_j$  is the RTT of path  $j$ ,  $f_j$  is the fraction of packets on path  $j$ . The retransmission timeout (RTO) period for the  $i$ -th packet is  $RTO_i$ . According to [55], the current estimated RTT,  $R_i$ , can be calculated approximately as

$$R_i = \sum_{j=1}^n f_j r_j,$$

and

$$RTO_i = R + 4D,$$

where  $R$  and  $D$  are smoothed RTT and RTT deviation. The packet on path  $j$  will experience timeout permanently if  $r_j > RTO_i$ . Once a timeout event is triggered, according to its congestion control mechanism, the TCP congestion window will be set to one with slow start, which underutilizes the available network resource seriously and brings extra bandwidth fluctuation.

2. Packets may arrive at destination out of order, i.e., the packet receiving sequence in a flow is different from its sending sequence. As shown in Fig. 3.2, in the 2-path scenario, if packets 2, 3, and 4 arrive earlier than packets 1, the receiver will buffer the first arrived three out-of-order



packets and sent back duplicate ACKs. One the sender receivers 3 duplicate ACKs, it has to retransmit packet 1. But in fact, packet 1 is not missing. This unnecessary retransmission not only wastes network bandwidth, but also makes the utilization of available network resource low and introduces additional bandwidth variability as well.

3. The interference among multiple paths in WMNs further aggravates the degradation and fluctuation. Different from wired networks, in a WMN, even if different paths do not share common links or nodes, simultaneous transmissions along these paths would still suffer from serious interferences given the broadcast nature of wireless media.

### **3.3 Summary**

This chapter has presented an investigation of the characteristics of multiple path TCP in multi-channel WMNs. Through examining the simulation results, we have demonstrate that, with a naive traffic split, TCP could suffer from performance degradation and fluctuation. With a closer look, we observed that TCP was designed to delivery packet in order along a single path. Unfortunately, multiple path routing would bring persistent timeout events, packet reordering and wireless interference, which influence TCP performance severely. In the next chapter, we will propose an efficient scheme to address these challenges.

## Chapter 4

# Path Diversified Retransmission (PDR) and System Design

In this section, we present the Path Diversified Retransmission scheme (PDR), and analyze how it eliminates the persistent timeout as well as packet reordering problems and thus improves TCP performance with multiple path routing.

### 4.1 PDR

Different from the conventional even traffic split, our PDR differentiates the original data packets and the retransmitted packets, and delivers them over two separated paths with different optimization criteria. Specifically, as illustrated in Fig. 4.1, the number of the original packets is in general dominating, and thus they should be scheduled over the maximum-bandwidth path. On the other hand, the retransmitted packets are of relatively smaller amount, but expected to be delivered more

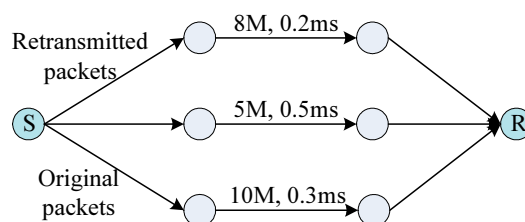


Figure 4.1: A network with 8 nodes. The available bandwidth and the end-to-end delay of each path are shown along each path.

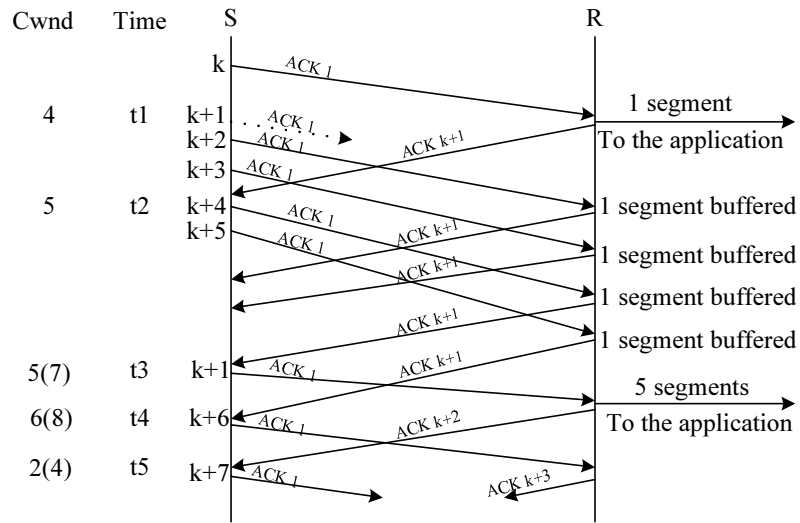
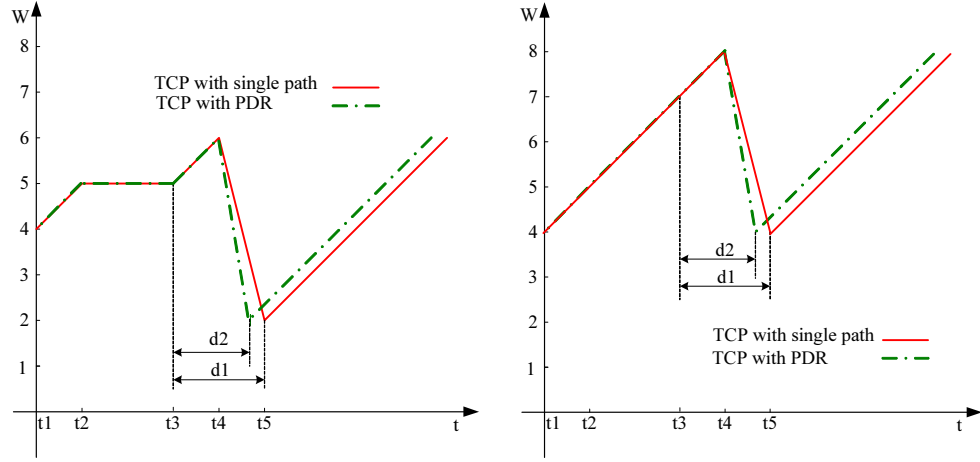


Figure 4.2: TCP loss recovery process in single path routing

quickly. Hence, they will be scheduled over the path with the minimum end-to-end delay. Here we assume that these two paths are disjoint and we will discuss the impact of correlations between them in Chapter 5.

PDR can reduce interference among concurrent multipath transmissions. On one hand, the missing packets are much less than the original packets. On the other hand, the arrival pattern of the retransmitted packets is not as continuous as that of the original packets. Thus the interference between the retransmitted packets and the original packets along the two paths is much less severe.

Intuitively, the PDR will not introduce extra packet reordering as other multipath protocols do. It will not underestimate the RTT, either, given the Karn's algorithm [37] in TCP does not count retransmitted segments in RTT estimation. In other words, the RTT estimation in PDR is the same as that in a single path case. Moreover, since the path delay for the retransmitted packets is smaller than that for the original packets, the retransmitted packets will seldom suffer from further timeout. These together improve the responsiveness and throughput of TCP, and better end-to-end performance can therefore be expected. To validate the effectiveness of PDR for TCP, we next analyze the improvement of TCP throughput with PDR.



(a) Evolution of congestion window size during congestion loss for TCP Reno in Fig. 4.2

(b) Evolution of congestion window size during random loss for TCP Reno in Fig. 4.2

Figure 4.3:  $d_1$  and  $d_2$  are the loss recovery times of TCP over a single path and that with PDR, respectively. If the retransmitted packets are sent on the path with minimum delay, we have  $d_2 < d_1$ .

## 4.2 Improvement of TCP Throughput with PDR

Consider an example in Fig. 4.2. At first (time  $t_1$ ), packet  $k + 1$  is lost and congestion window size is 4. The receiver buffers the packets with higher sequence numbers and returns duplicate ACKs. At time  $t_3$ , three duplicate ACKs are received at the sender side, and packet  $k + 1$  is therefore retransmitted. Meanwhile, according to the TCP Reno congestion avoidance algorithm [66], the threshold  $ssthresh$  is set to 2 and the congestion window size is changed to 5, which is one half of the current congestion window size, 5, plus 3. In TCP Reno, if the loss is triggered by congestion, the congestion avoidance algorithm is the same as that of TCP Reno; otherwise, the threshold  $ssthresh$  is set to 4 and the congestion window size is changed to 7. Then the congestion window increases linearly until a new ACK is received at time  $t_5$  and the congestion window size is set to  $ssthresh$ . The evolution of the congestion windows size is depicted in Fig. 4.3. We define the *loss recovery time* as the time between retransmitting the missing packet and receiving a new ACK. PDR delivers the retransmitted packets along the minimum-delay path, and hence its recovery time is smaller than that of the single path case (see dashdotted lines in Fig. 4.3). Meanwhile, PDR benefits from bandwidth aggregation in multipath routing, and therefore enhances TCP throughput, too.

We now formally evaluate the performance gain of PDR. Our analysis extends the TCP Reno throughput model [88] to the multipath scenario. The Reno's model itself is a modification of the

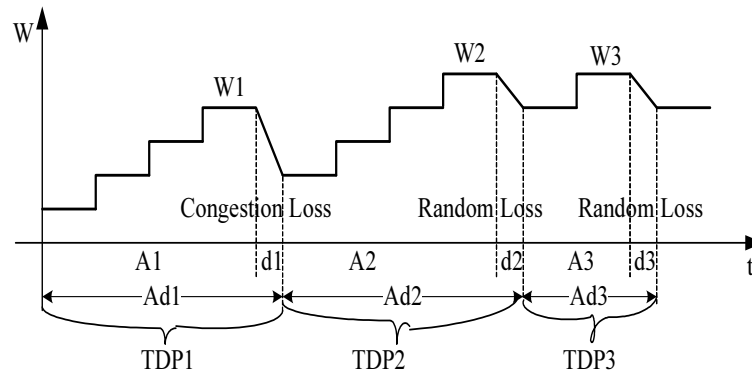


Figure 4.4: Evolution of window size over time when retransmission is triggered by three-duplicate ACKs

classical TCP Reno throughput model to the wireless network scenario [53]. Their key notations are summarized as follows:

- $TDP_i$ : interval between two ‘triple-duplicate’ ACK (TD) loss indications;
- $Ad_i$ : duration of  $TDP_i$ ;
- $A_i$ : duration of  $TDP_i$  without loss recovery time;
- $d_i$ : loss recovery time of  $TDP_i$ ;
- $Y_i$ : number of packets sent in  $TDP_i$ ;
- $r$ : RTT;
- $\gamma$ : the proportion of cwnd value after cut down;
- $p$ : packet loss rate;
- $B$ : TCP throughput;
- $b$ : the number of packets acknowledged by a received ACK;
- $Q$ :  $\frac{1}{E[n]}$ , where  $\{n_i\}_i$  is an independent identical distribution (i.i.d) sequence of random variables;
- $Z^{TO}$ : duration of a sequence of time-outs.

We generalize the window evolution process (Fig. 1 in [88] and [53]) with loss recovery time between TDPs, as shown in Fig. 4.4. As such,  $E[Ad]$  can be expressed as

$$E[Ad] = E[A] + E[d]$$

where  $E[A]$  has the same formula as that in [53]. As defined earlier,  $E[d] = E[r]$ . Let  $\Delta d$  be the difference between TCP loss-recovery time in PDR and that with a single path routing. We can view  $\{\Delta d_i\}_i$  as a random sequence obeying i.i.d in  $[0, E[r]]$ . It follows that  $E[\Delta d] = \frac{1}{2}E[r]$ . Note that  $B = \frac{E[Y]}{E[Ad]}$  is the TCP goodput instead of throughput in PDR, because the packets on the path with the maximum bandwidth are all original ones. Let  $B_1$  be the TCP throughput in PDR and  $B_2$  be the TCP throughput in a single path routing. The throughput improvement is therefore

$$\begin{aligned} \frac{B_1 - B_2}{B_2} &= \frac{\frac{E[Y](1+p)}{E[A]+E[d_1]} - \frac{E[Y]}{E[A]+E[d_2]}}{\frac{E[Y]}{E[A]+E[d_2]}} \\ &> \frac{E[\Delta d]}{E[A] + E[d]} \end{aligned}$$

where

$$\frac{E[\Delta d]}{E[A] + E[d]} = \frac{\frac{1}{2}}{\frac{b(1-\gamma)+1}{2(1+\gamma)} + \sqrt{\frac{2b(1-\gamma)(1-p)}{(1+\gamma)p} + \left(\frac{b(1-\gamma)+1}{1+\gamma}\right)^2 + 2}}.$$

Similarly, given triple-duplicate ACKs and timeouts, the TCP throughput improvement can be expressed as

$$\frac{B_1 - B_2}{B_2} > \frac{E[\Delta d] + QE[\Delta d]}{E[A] + E[d] + Q(E[Z^{TO}] + E[d])}.$$

Here,  $E[Y]$ ,  $E[r]$ ,  $Q$  and  $E[Z^{TO}]$  all have the same formulas as in [88]. Therefore, as long as the delay for the path of the retransmitted path is shorter, the TCP throughput can be improved.

### 4.3 System Design

We now discuss the seamless integration between TCP and multipath routing. Our implementation adopts a cross-layer design, as illustrated in Fig. 4.5. We add a classifier between the transport layer and the IP layer, which distinguishes the retransmitted data from the original packets. In the IP layer, we design JOB, a QoS-aware multiple path routing protocol that extends the AOMDV protocol [47] to collaborate with the classifier. JOB discovers and maintains the maximum-bandwidth path and the minimum-delay path concurrently, so as to deliver the original packets and the retransmitted packets, respectively. Next, we elaborate the detailed implementation of the different modules.

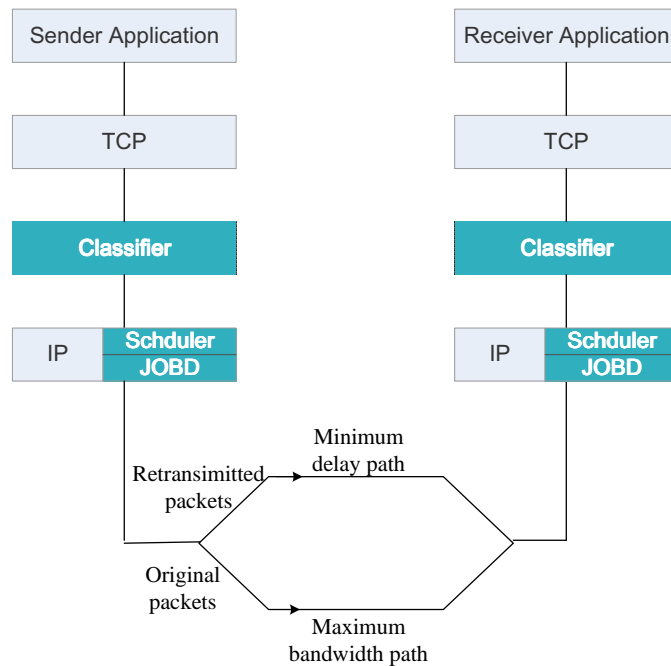


Figure 4.5: Overview of the cross-layer implementation

The classifier characterizes the original packets and the retransmitted packets by checking the *sequence number* field in the TCP header of each incoming packet. Specifically, it reserves the newest TCP sequence number. When receiving a packet from the transport layer, the classifier gets TCP sequence number from the packet header and compares it with the newest sequence number it maintains. If it is an old sequence number, the classifier marks it as a retransmitted packet. Otherwise, it designates an *original* label to the packet, and the classifier updates its newest sequence number.

The scheduler receives marked packets from the classifier and schedules them according to PDR. It keeps two queues, retransmitted queue and original queue, for the two paths, respectively. The retransmitted queue has higher priority since missing packets look forward to be delivered as quickly as possible.

We will discuss the design and implementation of JOB routing protocol in detail in the next chapter.

## 4.4 Summary

We have presented our PDR in this chapter, i.e., scheduling the retransmitted packets across the path with minimum delay while transmitting the original packets along the maximum-bandwidth path. Intuitively, it will not introduce additional packet reordering, persistent timeout events and reduce interference between concurrent two path transmission. Through extended analysis based on the TCP Reno throughput model, we indicate that our PDR will reduce loss recovery time so as to improve TCP throughput. After that, we have proposed a cross layer design for PDR being implemented. We added a classifier and a scheduler between the IP layer and the transport layer to mark the retransmitted packets from the original packets and schedule them to the appropriate paths.



## Chapter 5

# Joint Optimization of Bandwidth and Delay (JOBBD) Routing Protocol

In this chapter, we present the design and implementation detail of Joint Optimization of Bandwidth and Delay (JOBBD) routing protocol. First we give a general case study on finding multiple disjoint paths with diverse QoS characteristics, then we design new computing algorithms of bandwidth and delay under channel diversity with the consideration of intra-flow and inter-flow interference. After that, we describe some practical issues in the implementation of JOBBD.

### 5.1 System Overview and Problem Statement

In this section, we study on a general case that is discovering multi-QoS optimized disjoint paths in multiple channel WMNs. we first give a system overview and then state and solve the problem.

We consider a wireless mesh network of  $K$  nodes (wireless routers), in which any two nodes can communicate directly or through multi-hop relays. Each node is configured with  $N$  full-duplex wireless interfaces ( $N \geq 2$ ). There are also  $N$  orthogonal channels in the WMN and each interface is allotted with one channel.<sup>1</sup> We refer to each channel between two neighboring nodes as a *channel link*, or a *link* in short. Hence, there are  $N$  links between two neighboring nodes in our system, as illustrated in Fig. 5.1 where  $N = 2$ .

---

<sup>1</sup>These interfaces are identical with the same transmission and carrier sensing range, and can be switched from one channel to another within marginal delay. In practice the number of channels can be more than that of interfaces, and there have been a number of dynamic channel assignment algorithms available [10, 39, 67, 77].

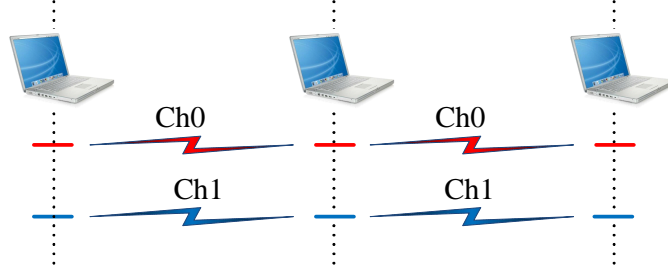


Figure 5.1: A multi-channel WMN with 3 nodes. Each node is configured with two channels, and thus there are two links between neighboring nodes.

The goal of our multi-channel multi-path routing protocol is to find two disjoint paths with different QoS-optimized performance. We are particularly interested in bandwidth and delay, although our solution framework is general in accommodating other QoS parameters.

We transform the WMN into a weighted undirected graph  $G(V, E)$  as shown in Fig. 5.2, where  $V$  is the set of nodes and  $E$  is the set of links. In particular,  $V$  consists of two kinds of nodes, namely, *master nodes* (or *M-nodes* in short) and *interface nodes* (or *I-nodes* in short), representing the nodes and their interfaces in the WMN, respectively. Let  $V_m$  denote the set of M-nodes, and  $V_i$  denote the set of I-nodes. Obviously, every M-node has  $N$  I-nodes and only the nodes in set  $V_m$  can be selected as source  $s$  and destination  $t$ . The edges are also categorized into two sets,  $E_m$  and  $E_i$ . The edges in  $E_m$  are between M-nodes and their I-nodes. The edges in  $E_i$  are from every I-node of a M-node to the I-node of other M-nodes if two interfaces of different nodes operate on the same channel and within each other's transmission range in the WMN. Every edge  $e$  in  $E$  has two costs, namely, its available bandwidth and delay. Let  $Bw(e)$  be the available bandwidth of  $e$ ,  $Delay(e)$  be the delay of  $e$ , and  $C(e)$  be the channel capacity of  $e$ . We have  $Bw(e) = \infty$  and  $Delay(e) = 0$  if  $e \in E_m$ , as interfaces are switchable with marginal latency in one node in the WMN. For  $e \in E_i$ ,  $0 \leq Bw(e) \leq C(e)$  and  $Delay(e) \geq 0$ .

The bandwidth- and delay-optimized disjoint path selection problem in multi-channel WMNs can then be formulated as follows:

Given network  $G(V, E)$ , a source node  $s$  and a destination node  $t$  ( $s, t \in V_m$ ), find two paths  $P_{s,t}^{bw}$  and  $P_{s,t}^{delay}$  such that

$$P_{s,t}^{bw} \cap P_{s,t}^{delay} = \{s, t\},$$

with objective function

$$\max(\min_{e \in P_{s,t}^{bw}} Bw(e))$$

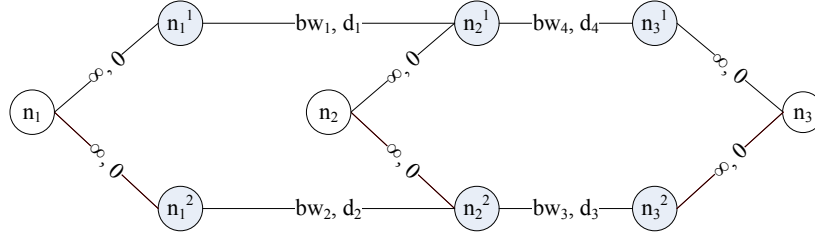


Figure 5.2: A multi-channel WMN with 3 nodes,  $n_1$ ,  $n_2$  and  $n_3$ . Each node is configured with two interfaces. Each interface is considered as an I-node and connected to its M-node by an edge with infinite bandwidth and zero delay. Bandwidth and delay on each edge are shown. We assume  $bw_1 > bw_2$ ,  $d_1 > d_2$ ,  $bw_3 > bw_4$ ,  $d_3 > d_4$ . If  $n_1$  is the source node and  $n_3$  is the destination, the maximum-bandwidth path will be  $n_1 \rightarrow n_1^1 \rightarrow n_2^1 \rightarrow n_2 \rightarrow n_2^2 \rightarrow n_3^2 \rightarrow n_3$  and the minimum-delay path will be  $n_1 \rightarrow n_1^2 \rightarrow n_2^2 \rightarrow n_2 \rightarrow n_2^1 \rightarrow n_3^1 \rightarrow n_3$ .

$$\min\left(\sum_{e \in P_{s,t}^{delay}} Delay(e)\right).$$

**Theorem 1.** *There exist complete disjoint paths with bandwidth- and delay-optimized performance, if the bandwidth- and delay-optimized channel links are different between neighboring nodes.*

*Proof.* If the the bandwidth- and delay-optimized paths do not share any neighboring node pairs, the two paths are clearly disjoint. Otherwise, for each overlapped neighboring node pair, the two paths can respectively pick up the bandwidth- and delay-optimized channel link between the two neighboring nodes and thus the two QoS-optimized paths are disjoint with no overlapped channel links. Note that node disjunction is not a requirement here, given the multiple interfaces can operate simultaneously in a node.  $\square$

Given the above theorem, the disjoint path selection problem can be simply transformed into the classical max-flow problem and the shortest-path problem, respectively, and then solved separately.

One concern here is that the bandwidth-optimized link would overlap with the delay-optimized link between neighboring nodes. We however find this is not necessarily the case. While a high bandwidth link has lower transmission delay, the total delay of a link depends on many other factors, in particular, the number of packets that have queued for transmitting over the channel. Existing works [45, 56] have suggested that the latter is often the dominating factor, which is also confirmed by our analysis (see Section 6.2). As such, the probability that the two channel links overlap can be quite low if there are a number of channels available in the network (for IEEE 802.11a standard, there are 12 orthogonal channels).

Table 5.1: List of Notations

$G(V, E)$	a graph with $V$ as its vertex set and $E$ as its edge set
$V_m$	set of M-nodes
$V_i$	set of I-nodes
$E_m$	set of edges between M-nodes and their I-nodes
$E_i$	set of edges between I-nodes
$Bw(e)$	available bandwidth of edge $e$
$C(e)$	channel capacity of edge $e$
$Delay(e)$	delay of edge $e$
$P_{s,t}^{bw}$	maximum-bandwidth path from node $s$ to $t$
$P_{s,t}^{delay}$	minimum-delay path from node $s$ to $t$
$CHB_i$	channel busy time ratio for link $i$
$ACB_i$	accumulated channel busy time for link $i$
$T_{total}$	total channel time slot
$B_i$	channel capacity of link $i$
$B_{ITF_i}$	available bandwidth of link $i$ in inter-flow interference
$ETX_i$	expected transmission count in link $i$
$TB_i$	channel busy time of link $i$
$TB_{1,2,\dots,k}$	accumulated channel busy time over links 1, 2, ..., and $k$
$B_{IAF}$	available bandwidth of a path in intra-flow interference
$D_p$	end-to-end delay over path $p$
$H$	hop count of path $p$
$D_i$	one hop delay of link $i$
$I$	packet length
$AB_i$	available bandwidth of link $i$
$W_j$	contention window at the $j$ th bankoff stage
$M_i$	number of packets queued in the buffer of link $i$

A more critical challenge however is the accurate estimation of bandwidth and delay in this multi-channel environment. It is known that co-channel interference is dominating when multiple channels are available, which unfortunately has not been well addressed by state-of-the-art estimation algorithms [43, 62]. In the next section, we will address this challenge and develop more accurate estimation tools under channel diversity, and we will further present a practical protocol design in Section 5.3. To facilitate our discussion, Table 5.1 summarizes the key notations used in this paper.

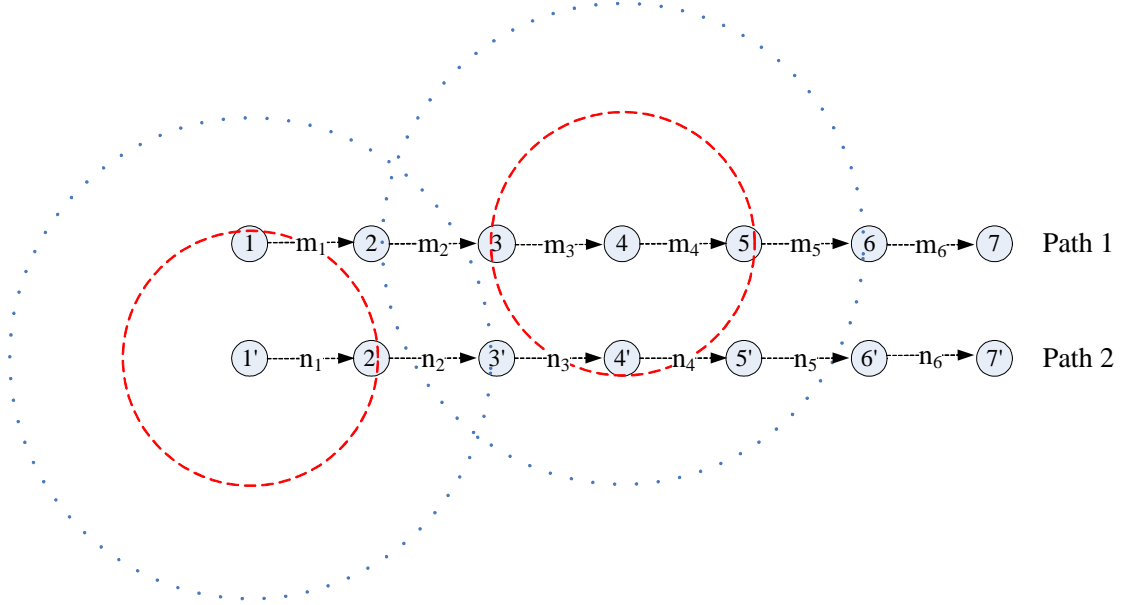


Figure 5.3: Example of inter-flow interference between two paths. Assume links  $m_2$ ,  $m_5$  and  $n_1$  are assigned with the same channel. The small circle covers available transmission range of a node, and the large circle denotes interference range of a node. So links  $m_2$  and  $n_1$  will suffer from co-channel inter-flow interference, while  $m_5$  and  $n_1$  have no such interference.

## 5.2 Bandwidth and Delay Estimation under Channel Diversity

We proceed with new algorithm designs for computing the two routing metrics in the multi-channel environment, i.e., available bandwidth and end-to-end delay of a path. It is known that, due to the limitation of available orthogonal channels, with the existence of *co-channel links*, i.e., the links operating on the same channel, both inter-flow interference and intra-flow interference can occur [83]. The former refers to the contentions that occur among co-channel links in different paths, and latter refers to the contentions when co-channel links locate in the same path. We will first compute the available bandwidth under inter-flow interference and intra-flow interference separately, and then integrate them together to calculate the available bandwidth over a whole path.

### 5.2.1 Available Bandwidth under Inter-Flow Interference

We compute available bandwidth of link  $i$  under inter-flow interference,  $B_{ITF_i}$ , as follows:

$$B_{ITF_i} = \frac{(1 - CHB_i)B_i}{ETX_i}, \quad (5.1)$$

---

**Algorithm I** Channel Busy Time for Individual Link  
under Inter-Flow Interference

---

Let  $TB_j$  be the channel busy time of link  $j$

Let  $S$  be the set of links that inter-flow interfere with link  $i$

Let  $ACB_i$  be the accumulated channel busy time for link  $i$ .

Divide  $S$  into a minimum number (denoted as  $m$ ) of subsets, where links have no interference with each other in every subset

but interfere with links in other subsets,

denoted as  $S_1, S_2, \dots, S_m$ .

**for**  $j = 1$  to  $m$

    Suppose there are  $k$  links in subset  $S_j$

$$ACB_{ij} = \frac{\sum_{l=1}^k TB_l + \max(TB_1, TB_2, \dots, TB_k)}{2}$$

**end for**

$$ACB_i = \sum_{j=1}^m ACB_{ij}$$


---

where  $B_i$  is channel capacity of link  $i$ ,  $CHB_i = ACB_i/T_{total}$  is *channel busy time ratio*,  $ACB_i$  is *accumulated channel busy time* of links inter-flow interfered with link  $i$ ,  $T_{total}$  is total channel time slot including both ‘busy’ and ‘idle’ periods, and  $ETX_i$  represents *expected transmission count* required to successfully deliver a packet along link  $i$ . Obviously,  $1 - CHB_i$  indicates channel idle ratio for link  $i$ , and the numerator part of  $B_{ITF_i}$  can be interpreted as the available bandwidth for one transmission. As such,  $B_{ITF_i}$  gives the effective available bandwidth for one successful transmission with  $ETX_i$  transmission attempts.

Note that the accumulated channel busy time,  $ACB_i$ , can not simply be the sum of all the channel busy time of links that suffer from inter-flow interferences with link  $i$ , since these links may have no interference with each other. If two co-channel links,  $m$  and  $n$ , do not locate in each other’s interference range (such as links  $m_5$  and  $n_1$  in Fig. 5.3), their accumulated channel busy time will be in the range of  $[\max(TB_m, TB_n), TB_m + TB_n]$  as  $TB_m$  and  $TB_n$  are independent with each other, where  $TB_m$  and  $TB_n$  are their channel busy times, respectively. And we can view their accumulated channel busy time  $TB_{mn}$  as a uniformly distributed random variable in  $[\max(TB_m, TB_n), TB_m + TB_n]$  approximately. It follows that  $E[TB_{mn}] = (TB_m + TB_n + \max(TB_m, TB_n))/2$ . Otherwise, if  $m$  and  $n$  interfere with each other (such as links  $m_2$  and  $n_1$  in Fig. 5.3),  $TB_{mn}$  can simply be the sum of their channel busy time, since  $TB_m$  and  $TB_n$  could not occur simultaneously. In general, if co-channel links  $l_1, l_2, \dots$ , and  $l_k$  have no interference with each other (note that these links all have co-channel interference with a common link), the accumulated channel busy time  $TB_{1,2,\dots,k}$

complies with uniform distribution in  $[\max(TB_1, TB_2, \dots, TB_k), \sum_{l=1}^k TB_l]$  approximately, and

$$E[TB_{1,\dots,k}] = \frac{\sum_{l=1}^k TB_l + \max(TB_1, TB_2, \dots, TB_k)}{2} \quad (5.2)$$

Therefore, we first divide the set of all links inter-flow interfering with link  $i$  into a minimum number of subsets, where the links do not interfere with each other in every subset but have interference with links in other subsets. For any subset  $S_j$ , the accumulated channel busy time  $ACB_{ij}$  is calculated according to Eq. (5.2), and the subset is viewed as a newly mixed virtual link with  $ACB_{ij}$  as its channel busy time. The overall accumulated channel busy time is the sum of the accumulated channel busy time of each subset, for there are interferences across links in different subsets. This calculation is summarized in Algorithm I. And in Algorithm I, the problem that how to divide  $S$  could be formulated as follows:

For  $L = \{TB_1, TB_2, \dots, TB_n\}$

Find  $S = \{S_1, S_2, \dots, S_m\}, m \leq n$

Such that,

(1)  $\forall TB_i \in L, \exists S_j \subset S, TB_i \in S_j$

(2)  $\cup S_j = L$

(3)  $\forall S_i, S_j \subset S, S_i \cap S_j = \emptyset$

(4)  $\forall S_i \subset S, \forall TB_i, TB_j \in S_i, TB_i$  and  $TB_j$  has no interference with each other.

Objective

$$\min \sum_{i=1}^m \frac{1}{2} [\max_{TB_j \in S_i} TB_j + \sum_{TB_j \in S_i} TB_j].$$

We could employ the general greedy algorithm to figure out the division way of  $S$ .

In real system, the interference relationship between two links would change from time to time because of certain dynamic factors in practice, such as changed topologies, dynamic channel assignment schemes as well as alternative routing path due to heavy traffic load. Nevertheless, WMN has a relatively static infrastructure, so here we could periodically collect and recalculate the channel busy time not so frequently and the overhead would not that much.

### 5.2.2 Available Bandwidth under Intra-Flow Interference

When there are two co-channel links that interfere with each other in the same path, the two links can not transmit data simultaneously. Hence, we have

$$\frac{1}{B_{ij}} = \frac{1}{B_i} + \frac{1}{B_j} \quad (5.3)$$

where link  $i$  and link  $j$  are two interfered co-channel links in the same path,  $B_i$  and  $B_j$  are bandwidth of each link, and  $B_{ij}$  is the total available bandwidth over the two links. For the scenario that co-channel links  $l_1, l_2, \dots$ , and  $l_k$  interfere with each other in the same path, we have

$$\frac{1}{B_{l_1, l_2, \dots, l_k}} = \frac{1}{B_{l_1}} + \frac{1}{B_{l_2}} + \dots + \frac{1}{B_{l_k}} \quad (5.4)$$

where  $B_{l_1, l_2, \dots, l_k}$  is the total available bandwidth over links  $l_1, l_2, \dots$ , and  $l_k$ .

As such, we can first find all the links that interfered with a particular link in one path. The available bandwidth of that link is then calculated by Eq. (5.4) and we then take the minimum available bandwidth among all links as the available bandwidth for that multi-channel path under intra-flow interference, as illustrated in Algorithm II. This is a centralized algorithm to be executed by the last node in one path. We further extend it to a distributed implementation, which considers two adjacent co-channel links at a time and thus can be executed hop-by-hop (see Algorithm III). Note that, in both algorithms, the co-channel links are sorted by their distances from the source, following the flow direction.

The path available bandwidth from Algorithm III can be slightly smaller than that from Algorithm II, since one link might be involved in Eq. (5.3) several times while actually should just be involved in Eq. (5.4) one time. Nevertheless, state-of-the-art channel assignment algorithms are quite effective in minimizing the number of co-channel interfered links, and hence the difference of the two algorithms is marginal. Our experience is that the number of co-channel links interfering with each other is hardly beyond two.

### 5.2.3 Available Bandwidth over a Multi-Channel Path

Taking both inter-flow and intra-flow interference into account, the available bandwidth over a multi-channel path can be evaluated by integrating Algorithm I and Algorithm III. Specifically, We replace the channel capacity  $PB_{ij}$  in Algorithm III with the available bandwidth of the  $j$ th link under inter-flow interference, which is obtained by Algorithm I and Eq. 5.1. The  $B_{IAF}$  given by Algorithm III then becomes the available bandwidth over the multi-channel path.

### 5.2.4 End-to-End Delay over a Multi-Channel Path

The end-to-end delay is not only determined by path bandwidth, but also depends on other factors such as MAC access delay, and queuing delay [45]. To capture these factors, we compute the end-to-end delay of a path  $p$  as follows,



---

**Algorithm II** Centralized Algorithm for Available Bandwidth  
under Intra-Flow Interference

---

Let  $m$  be the number of different channels in one path  
 Let  $S_i$  be the set of links using channel  $i$   
 Let  $S'$  be an empty set  
 Let  $B_{IAF_{ij}}$  be the available bandwidth for link  $j$  using channel  $i$   
 Let  $B_{IAF_i}$  be the available bandwidth for the links using channel  $i$   
 Let  $B_{IAF}$  be the available bandwidth over a multi-channel path under intra-flow interference  
**for**  $i=1$  **to**  $m$   
      $S'_i = S_i$   
     **for** link  $j$  in  $S_i$  with the smallest hopcount from the sender  
         Suppose there are  $p$  links in set  $S'_j$  interfere with  $j$ ,  
         denoted as  $l_1, l_2, \dots, l_p$   
         
$$\frac{1}{B_{IAF_{ij}}} = \frac{1}{B_{l_1}} + \frac{1}{B_{l_2}} + \dots + \frac{1}{B_{l_p}} + \frac{1}{B_j}$$
  
         Remove link  $j$  from  $S_i$   
      $B_{IAF_i} = \min(B_{IAF_{i1}}, B_{IAF_{ij}})$   
     **end for**  
      $B_{IAF} = \min(B_{IAF_1}, B_{IAF_i})$   
**end for**

---



---

**Algorithm III** Distributed Algorithm for Available Bandwidth  
under Intra-Flow Interference

---

Let  $m$  be the number of different channels in one path  
 Let  $n_i$  be the number of links using channel  $i$   
 Let  $PB_{ij}$  be the channel capacity for the  $j$ th link under channel  $i$   
 Let  $B_{IAF_i}$  be the available bandwidth for the links using channel  $i$   
 Let  $B_{IAF}$  be the available bandwidth over a multi-channel path under intra-flow interference  
**for**  $i=1$  **to**  $m$   
     **for**  $j=1$  **to**  $n_i$   
         **if** two adjacent co-channel links within interference range  
             
$$\frac{1}{B_{IAF_i}} = \frac{1}{B_{IAF_i}} + \frac{1}{PB_{ij}}$$
  
         **else**  
              $B_{IAF_i} = \min(B_{IAF_i}, PB_{ij})$   
         **end if**  
     **end for**  
      $B_{IAF} = \min(B_{IAF_1}, B_{IAF_i})$   
**end for**

---

$$D_p = \sum_{i=1}^H D_i,$$

with

$$D_i = (ETX_i \frac{I}{AB_i} + \sum_{j=1}^{ETX_i} E[W_j]) \cdot M_i \quad (5.5)$$

The parameters here are summarized as follows, and the practical estimation of the key parameters will be detailed in the next section:

- $D_p$ : end-to-end delay over path  $p$ ;
- $H$ : hop count of path  $p$ ;
- $D_i$ : one hop delay in link  $i$ ;
- $I$ : packet length;
- $ETX_i$ : expected transmission count along link  $i$ ;
- $AB_i$ : available bandwidth of channel link  $i$ ;
- $W_j$ : contention window at the  $j$ th backoff stage;
- $M_i$ : number of packets queued in the buffer of link  $i$ .

It is worth noting that in Eq. (5.5),  $AB_i$  is the available bandwidth of channel link  $i$ , which is mainly influenced by co-channel interferences and different from that in single-channel networks. The computation of  $AB_i$  adopts the algorithms proposed in the previous subsection. Specifically, we first find all links inter-flow interfered with link  $i$ , and obtain the available bandwidth of link  $i$  under inter-flow interference according to Eq. (5.1). We then discover co-channel links interfered with link  $i$  in path  $p$ , and get  $AB_i$  through Eq. (5.4). According to the 802.11 standard [2, 19],  $E[W_j] = (2^{j-1}W_{min} - 1)/2$ , where  $W_{min}$  is the minimum contention window.

Intuitively, in Eq. (5.5),  $ETX_i \frac{I}{AB_i}$  denotes the transmission delay under channel diversity,  $\sum_{j=1}^{ETX_i} E[W_j]$  represents the MAC access delay, and  $M_i$  captures the queuing delay. Note that  $AB_i$  is the available bandwidth of channel link  $i$  rather than the available bandwidth of the whole path  $p$ . Comparing Algorithm III with Eq. (5.5), we can easily find that the delay is not simply inversely proportional to the available bandwidth of one path.

### 5.3 JOBDD Routing Protocol Design

In this subsection, we present a practical protocol design, JOBDD (Joint Optimization of Bandwidth and Delay), that incorporates the above results for discovering and utilizing the maximum-bandwidth path and the minimum-delay path simultaneously.

#### 5.3.1 Multi-Channel AOMDDV

##### Basic AOMDDV

Our JOBDD protocol inherits the route discovery mechanism from the Ad hoc On-Demand Multipath Distance Vector (AOMDDV) protocol [47], a well-known multipath extension to the classical AODV. AOMDDV computes multiple loop-free and link-disjoint paths with customized flooding, and takes hopcount as its routing metric. In AOMDDV, a node records the maximum hopcount of the multiple paths for each destination, referred to as the *advertised hopcount* for that destination. The protocol only picks up alternative routes with hopcount less than the *advertised hopcount*. The *first hop* field in an AOMDDV Route REQuest packet (RREQ) or Route REPLY packet (RREP), which indicates the first hop taken by the packet, is used to ensure the disjunction of the multiple paths.

##### Multi-Channel AOMDDV (MC-AOMDDV)

As described above, AOMDDV could serve as a single-channel routing protocol. To extend AOMDDV to support multi-channel transmission, we employ the scheme proposed by P. Kyasanur et al. [39], which exploits multiple channels with the employment of a multi-interface solution. Its main idea is to divide all the interfaces of one node into two groups, fixed interfaces and switchable interfaces, which is to handle the case that the number of available interfaces is no more than that of orthogonal channels. The fixed interfaces are for receiving packets sent to that node and the switchable interfaces are to transmit packets from that node. In other words, the channel for transmission is determined by the receiver, and the sender needs to adapt its switchable interface accordingly.

Besides the common routing tasks in single-channel network, MC-AOMDDV is mainly in charge of interface switching. We let each node maintain two datasheets, Neighbor Table (NT) and Channel Usage List (CUL). The NT records the fixed channels selected by its neighbors, indexed by IP. The CUL keeps the utilization of each channel, indexed by channel number. NT is added to its RREP and RREQ packets for assisting route discovery and maintenance. MC-AOMDDV adjusts its *switchable interface* complying with the *fixed interface* of node in the next hop according to its NT. Also, nodes

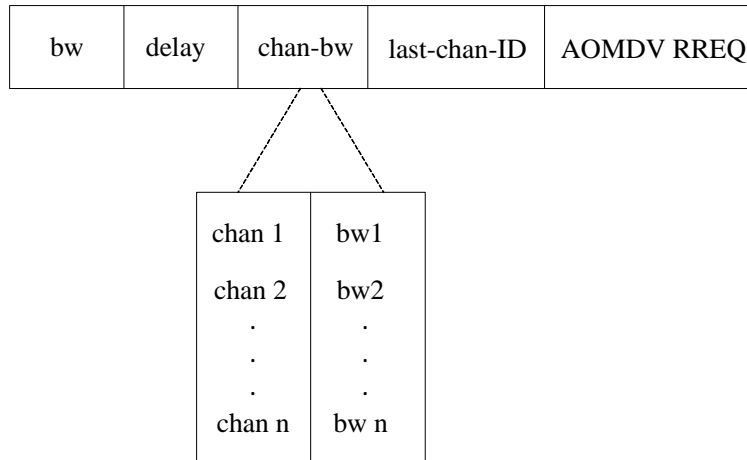


Figure 5.4: JOBBD RREQ packet header

could change their *fixed\_interface* dynamically, preferring the channel with smallest utilization.

### 5.3.2 Route Discovery in JOBBD

In our JOBBD protocol, when a node has data to transmit, it first broadcasts a *route request* message (RREQ) through all its interfaces. To discover QoS-optimized routes, we extend the header of each RREQ packet to  $\langle bw, delay, chan-bw, last-chan-ID, AOMDV\ RREQ\ header \rangle$ , as shown in Fig. 5.4. Here, *bw* and *delay* is the available bandwidth and end-to-end delay of the path passed by this RREQ, *chan-bw* records the available bandwidth for each channel, and *last-chan-ID* indicates last channel it passes. When one interface of an intermediate node receives an RREQ message at the first time, the node first gets the value of its channel (channel number is *last-chan-ID*) available bandwidth and one hop delay. It then updates *chan-bw*, the bandwidth of the channel. The *bw* is determined by the minimum bandwidth among all channel recorded in *chan-bw*. The *delay* field is updated similar as that of *hopcount* in the AOMDV header.

The node then rebroadcasts the message through all of its interfaces. Redundant RREQ messages can be used to build multiple reverse paths. Once the destination receives an RREQ packet, it updates the *bw* and *delay* fields as intermediate nodes, establishes the reverse path, and sends back RREP along the newly built reverse path. Subsequently, the destination or the nodes that have routes to the destination send back RREP packets with newly modified packet header,  $\langle bw, delay, last-chan-ID, AOMDV\ RREP\ header \rangle$ , to assist establishing forwarding paths. The three new fields in RREP have the same meanings as those in RREQ packet header.

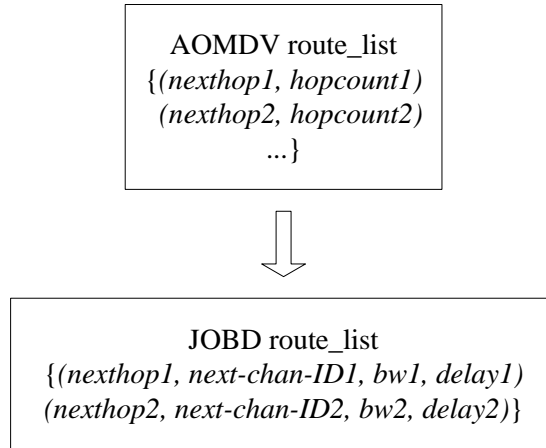


Figure 5.5: The route list of AOMDV and JOBDD

The routing table entries are also redesigned in JOBDD. Specifically, we add *next-chan-ID*, *bw* and *delay* fields in *route\_list*, as illustrated in Fig. 5.5. These three new fields indicate the channel number of next hop, the available bandwidth, and the end-to-end delay of one route, respectively. Similar as AOMDV, a route is updated when a node receives a RREQ or RREP packet with a new *sequence number* or there's a better path, e.g., with larger achievable bandwidth or smaller end-to-end delay.

### 5.3.3 Parameter Estimation

A key step toward implementing JOBDD is the estimation of the two path selection metrics, namely, available bandwidth and end-to-end delay. Given the algorithms in Section 5.2, this can be accomplished by measuring *ETX*, channel busy time, and the number of packets queued in the buffer.

To calculate *ETX*, we need to obtain the forward and reverse delivery ratio ( $d_f$  and  $d_r$ ). The value of  $d_f$  and  $d_r$  can be evaluated through link probing [20]. Each node periodically broadcasts probing packets, say, at a rate of one packet per second. Every node records the number of probing packets it receives during last ten seconds and inserts this information in the header of its own probe packets. Therefore, the nodes can compute  $d_r$  directly from the number of packets they receive during a recent period, say, ten seconds, and also obtain  $d_f$  by using the information in the probe packets sent to themselves from one of their neighbors.

For the measurement of channel busy time, we take the time that packets are sent successfully from one node along one channel as the busy time of that channel in a certain time period.

The number of packets queued in the buffer can also be estimated through probing packets [43]. Specifically, each node periodically broadcasts probing packets to its downstream neighbors at a predetermined rate, carrying over the number of packets queued in the buffer in its probe packet header. When downstream nodes receive the probing packets, they will update the count of packets queued in their upstreaming nodes in their neighbor lists.

## 5.4 Summary

In this chapter, we have described the design of JOB D in detail. We have shown that, as long as the best channels with different QoS metrics are not overlapped between WMN node pairs, there exist complete disjoint paths with different QoS metrics. We presented efficient solutions to discover such paths, particularly for bandwidth- and delay-optimization. We further designed new computing tools to calculate available bandwidth and end-to-end delay under channel diversity, considering both inter-flow interference and intra-flow interference. We also elaborated detailed implementation of JOB D, including route discovery and parameter estimation.

## Chapter 6

# Performance Evaluation

In this section, we evaluate the TCP performance with PDR. We also compare it with single path transmission (AODV) and conventional multipath transmission (AOMDV), denoted as nPDRs and nPDRm, respectively. We are particularly interested in the following performance measures:

*Throughput.* The TCP throughput is the average number of packets all destinations receive during a time unit.

*Fairness index.* The fairness index is to measure friendliness among the TCP flows, and Jain's fairness index [33] has been widely adopted in the literature.

We summarize our main observations as follows.

- PDR constantly outperforms nPDRs and nPDRm in terms of TCP throughput;
- PDR does not noticeably affect TCP friendliness, and its fairness index is similar to that without multipath transmission.

In the following subsections, we will first examine the probability that bandwidth- and delay-optimized paths are overlapping under different network sizes and different number of channels. We then present the experimental results which show that TCP-based applications can gain noticeable benefits from our solution.

### 6.1 Simulation Environment

Besides the same scenarios used in Chapter 3, we also simulated a much larger network with 100 nodes, distributed in a  $10 \times 10$  grid topology. The distance between adjacent nodes is 200m. Other

configurations are the same as that of the 40-node random network. We again performed 30 runs, with 10 TCP flows between randomly chosen sources and destinations in each run.

## 6.2 Simulation Results

### 6.2.1 Probability of Path Overlapping

As discussed earlier, the effectiveness of our solution largely depends on whether the maximum-bandwidth and minimum-delay channel links are overlapped at each hop. Suppose we have  $N$  orthogonal channels. For a neighboring node pair, the probability that these two overlap is  $1/N$ . For a path of  $L$  hops, the probability that an overlap occurs is  $1 - (1 - 1/N)^L$ . For a reasonably large  $N$ , say 12, this probability is pretty low. To better understand this, we have also measured this probability in our simulations. We vary the network size from 20 to 60 in the random topologies, and from 80 to 120 in the grid topology. In each topology, the amount of channels is changed from 8 to 16 and we randomly select up to 300 node pairs as sources and destinations.

Fig. 6.1(a) shows how the probability of path overlapping changes with different number of channels and nodes in random topology. Not surprisingly, it decreases with increasing the number of channels, as more channels potentially offer more disjoint channel links. It increases with increasing the network size, as there are more longer paths in a larger network. Nevertheless, it remains pretty low in all the settings (no more than 6%).

The results of grid topology are shown in Fig. 6.1(b), which follow a similar trend as in the random topologies. An interesting observation is that when the network is larger than a certain size (over 90 in our simulations), the further increase of the overlapping probability becomes marginal. A close investigation shows that at this stage, although the paths in the network may become longer, there are indeed more paths available between node pairs, i.e., paths are more diversified. We thus believe that in practice, the overlapping probability is still acceptable even with very large network sizes.

### 6.2.2 Throughput Results

In Fig. 6.2, we show the TCP throughput under different traffic and network configurations. The first three subfigures are for the 40-node random network. It can be seen that PDR generally outperforms nPDRs, and the improvement ranges from about 10% to 25%. Fig. 6.2(d) compares the TCP throughputs under PDR, nPDRs and nPDRm in the 100-node grid network. It again confirms that



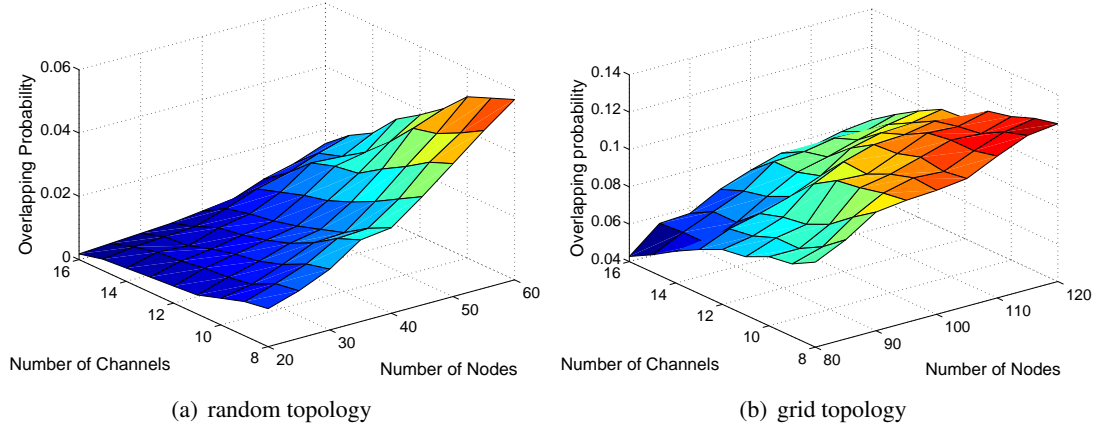


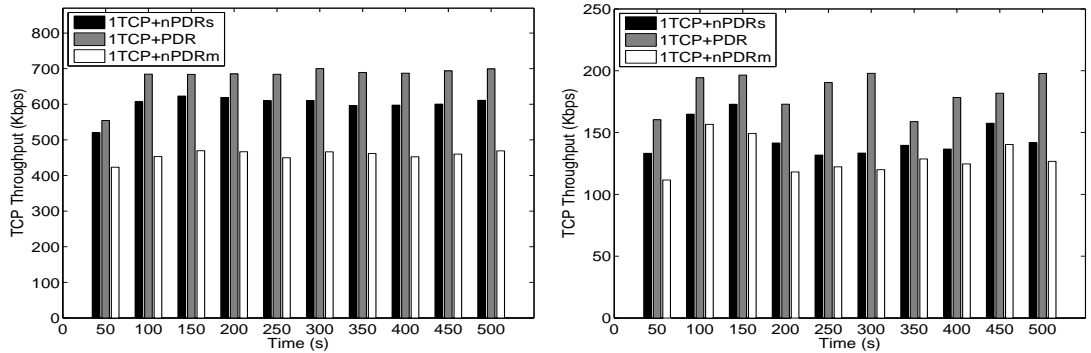
Figure 6.1: Probability that the bandwidth- and delay- optimized paths are overlapping

PDR enhances the TCP throughput. Through analyzing the data traces, we find that three factors contribute to such improvement: (1) Path diversity, which inherently amplifies the path bandwidth through aggregation; (2) Faster retransmission, which is achieved through selecting the low-delay path for retransmission; and (3) No extra packet reordering and timeout events are introduced.

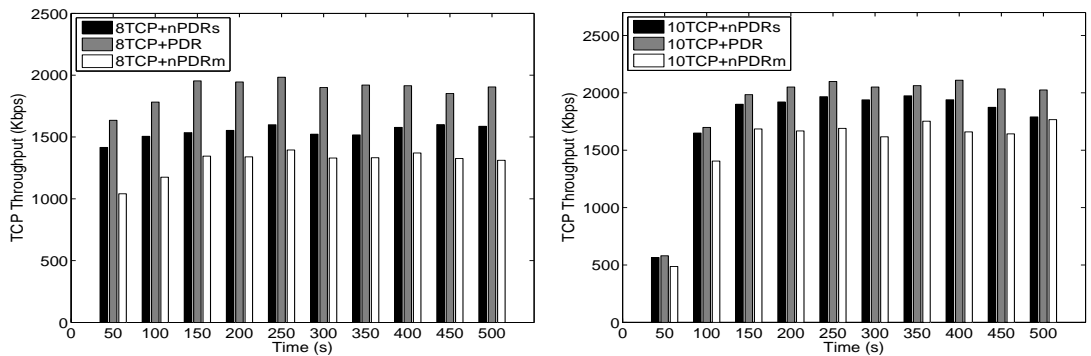
Note that the throughput improvement in the 100-node grid network is not as much as that in the 40-node random network. We find that this is mainly due to the special grid layout of the network. In this layout, if the sender and the receiver are in the same line, it is probable that the paths for the original packets and the retransmitted packets in PDR are the same, which thus brings no benefits with PDR.

Also note that TCP throughput at 50s is noticeably lower than that in other time instances in Fig. 6.2. This is because TCP takes time to reach an equilibrium. In addition, the more flows compete in networks, the slower TCP will attain its stability, which can be observed by comparing the first three subfigures with the last one. The TCP throughput also fluctuates more often when UDP flows exist.

To better illustrate the TCP throughput fluctuation, we show the detailed TCP throughput over time in Fig. 6.3. We can see that the variability of TCP throughput with nPDRm is the worst. This is again due to the frequent spurious retransmissions caused by packet reordering and timeout. The more false congestion controls are triggered, the more frequent TCP throughput fluctuates. Our PDR however is much more stable.



(a) TCP throughput (1 TCP flow in a 40-node random network). The average TCP throughputs for nPDRs, PDR, and nPDRm are 599.6Kbps, 676.3Kbps, and 457.2Kbps, respectively. (b) TCP throughput (1 TCP and 7 UDP flows in a 40-node random network). The average TCP throughputs for nPDRs, PDR, and nPDRm are 147.3Kbps, 185.9Kbps, and 125.8Kbps, respectively.



(c) TCP throughput (8 TCP flows in a 40-node random network). The average TCP throughputs for nPDRs, PDR, and nPDRm are 1341.2Kbps, 1639.1Kbps, and 1296.5Kbps, respectively. (d) TCP throughput (10 TCP flows in a 100-node grid network). The average TCP throughputs for nPDRs, PDR, and nPDRm are 1751.9Kbps, 1869.6Kbps, and 1537.5Kbps, respectively.

Figure 6.2: Comparisons of TCP throughput

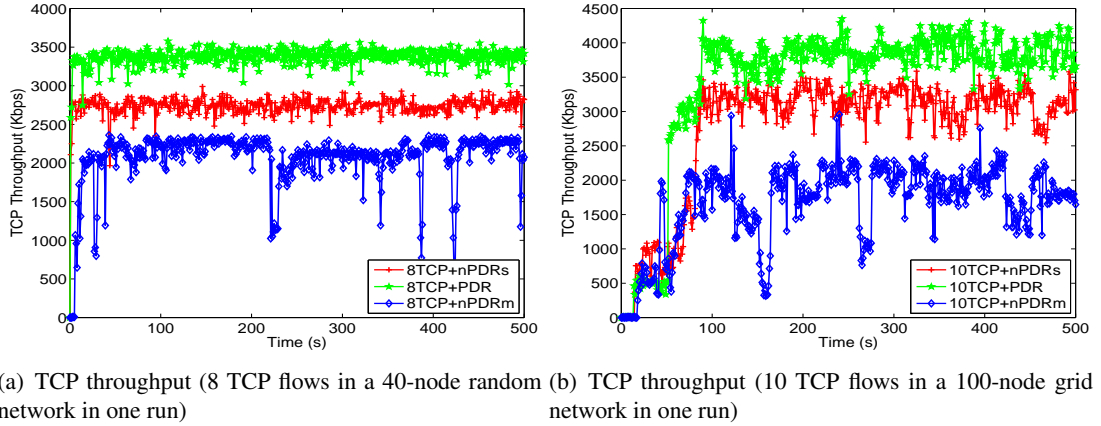
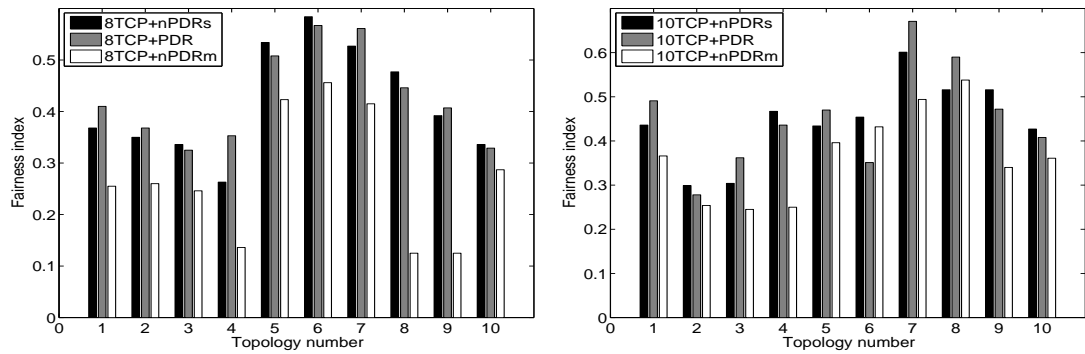


Figure 6.3: Comparisons of TCP throughput fluctuation

### 6.2.3 Fairness Results

We next show the fairness index in Fig. 6.4. We observe PDR has similar average fairness index as that of nPDRs, which implies that our solution will not affect TCP fairness under path diversity. On the other hand, nPDRm has lower fairness index, and we believe that it is due to the excessive reordering of packets and the timeout events. Consider an extreme scenario where the re-ordered packets are all from the same TCP flow, the throughput of this flow will be degraded significantly while the throughput of other flows are not affected at all, resulting in serious unfairness among TCP flows. On the other hand, PDR does not have such problem since every flow has retransmitted packets and the possibility of retransmission is equal across the flows.

The insight reason is that improper load splitting may result in large RTT differences among TCP flows and therefore brings more unfairness. TCP ensures fairness through its Additive Increase and Multiplicative Decrease (AIMD) congestion control, which is tightly coupled with RTT estimation. As a result, TCP throughput changes with RTT variations. The larger differences of RTTs exist among the flows, the less fairness TCP offers. The variations of fairness indexes in different topologies are also likely due to the RTT differences, because the average distance between sender and receiver pairs differs with topology.



(a) Fairness index (8 TCP flows in a 40-node random network and only one run for each topology). The average fairness indexes for nPDRs, PDR and nPDRm are 0.431, 0.442, and 0.273, respectively. (b) Fairness index (10 TCP flows in a 100-node grid network and only one run for each topology). The average fairness indexes for nPDRs, PDR and nPDRm are 0.453, and 0.328, respectively.

Figure 6.4: Comparisons of TCP fairness index

### 6.3 Summary

We have performed extensive simulations, and presented the comparisons in this chapter. The results demonstrate that our design greatly increases performance of TCP throughput and has no influence on TCP fairness, and we believe this is due to that PDR will not introduce additional packet reordering and timeout events.

## Chapter 7

# Conclusion and Future Work

### 7.1 Conclusion

In this paper, we have presented PDR, a retransmission scheme employing QoS-aware multi-path routing for TCP-based applications in WMNs. PDR maps the original packets and the retransmitted packets to different paths according to their distinctive QoS requirements. To cooperate with this scheme, we have designed JOBD routing protocol, which extends the conventional AOMDV by establishing distinct paths with the maximum bandwidth and the minimum delay, respectively. Our PDR does not trigger spurious retransmissions caused by reordering of packets and RTT underestimation. Therefore, different from conventional multipath transmission, it can enjoy the benefits from path diversity and meanwhile does not introduce additional bandwidth degradation and fluctuation. This has been validated by both theoretical analysis and simulation results.

### 7.2 Future Work

PDR is easy to be implemented in terms of traffic splitting. However, it may not achieve the maximum utilization of network resource as we just simply schedule the retransmitted packets to the minimum-delay path, with no considerations of the relationship between the packet loss rate and the available bandwidth on the minimum-delay path. Specially, if the packet loss rate is large while the available bandwidth is small, the retransmitted packets maybe arrive later as compared to those along the maximum-bandwidth path. Therefore, in the future, we plan to design dynamic assignment for the retransmitted packets and

# Bibliography

- [1] <http://www.kernel.org/>.
- [2] Wireless lan medium access control (MAC) and physical layer (PHY) specifications, 1999. ANSI/IEEE Std 802.11: 1999 (E) Part 11, ISO/IEC 8802-11.
- [3] I. F. Akyildiz and X. Wang. Cross-layer design in wireless mesh networks. *IEEE Transactions on Vehicular Technology*, 57(2):1061–1076, 2008.
- [4] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks and ISDN Systems*, 47(4):445–487, 2005.
- [5] I.F. Akyildiz and X. Wang. A survey on wireless mesh networks. *IEEE Communications Magazine*, 43(9):S23–S30, 2005.
- [6] Y. Amir, C. Danilov, M. A. Kaplan, R. Musaloiu-Elefteri, and N. Rivera. On redundant multipath operating system support for wireless mesh networks. In *Proceedings of IEEE SECON*, June 2008.
- [7] Y. Andreopoulos, N. Mastronarde, and M. Schaar. Cross-layer optimized video streaming over wireless multihop mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(11), 2006.
- [8] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee. On multiple description streaming with content delivery networks. In *Proceedings of IEEE INFOCOM*, 2002.
- [9] J. G. Apostolopoulos and M. D. Trott. Path diversity for enhanced media streaming. *IEEE Communications Magazine*, 42(8):80–87, 2004.
- [10] S. Avallone, I. F. Akyildiz, and G. Ventre. A channel and rate assignment algorithm and a layer-2.5 forwarding paradigm for multi-radio wireless mesh networks. *IEEE/ACM Transactions on Networking*, 17(1):267–280, 2009.
- [11] C. Barakat and E. Altman. Bandwidth trade-off between TCP and link-level FEC. *Computer Networks*, 39(2):133–150, 2002.
- [12] J. Bennett, C. Partridge, and N. Shectman. Packet reordering is not pathological network behavior. *IEEE/ACM Transaction on networking*, 7(6):789–798, 1999.

- [13] E. Blanton and M. Allman. On making TCP more robust to packet reordering. *ACM SIGCOMM Computer Comm. Rev.*, 32(1):20–30, 2002.
- [14] S. Bohacek, P. Hespanha, J. Leeand, C. Lim, and K. Obraczka. A new TCP for persistent packet reordering. *IEEE/ACM Transaction on networking*, 14(2):369–382, 2006.
- [15] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sansadidi, and R. Wang. TCP Westwood: end-to-end congestion control for wired/wireless networks. *Wireless Networks*, 8(5):467–479, 2002.
- [16] J. C. Chen, S. H. Chan, and V. Li. Multipath routing for video delivery over bandwidth-limited networks. *IEEE Journal on Selected Areas in Communications*, 22(10):1920–1932, 2004.
- [17] L. Chen and W. B. Heinzelman. QoS-aware routing based on bandwidth estimation for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(3):561–572, 2005.
- [18] R. Cheng and H. Lin. A cross-layer design for TCP end-to-end performance improvement in multi-hop wireless networks. *Computer Communication*, 31(14):3145–3152, 2008.
- [19] Y. Cheng, X. Ling, W. Song, L.X. Cai, W. Zhuang, and X. Shen. A cross-layer approach for WLAN voice capacity planning. *IEEE Journal on Selected Areas in Communications*, 25(4):678–688, 2007.
- [20] D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high- throughput path metric for multi-hop wireless routing. In *Proceedings of ACM MOBICOM*, Sep. 2003.
- [21] P. Cuetos and K. W. Ross. Adaptive rate control for streaming stored fine-grained scalable video. In *Proceedings of ACM NOSSDAV*, 2002.
- [22] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of ACM MOBICOM*, Sep. 2004.
- [23] A. Ford, C. Raiciu, and M. Handley. TCP extensions for multipath operation with multiple addresses, Oct. 2009. <http://tools.ietf.org/html/draft-ford-mptcp-multiaddressed-02>.
- [24] Cheng Peng Fu and S. C. Liew. TCP Veno: TCP enhancement for transmission over wireless access networks. *IEEE Journal on Selected Areas in Communications*, 21(2):216–228, 2003.
- [25] C.P. Fu and S.C. Liew. Tcp veno: Tcp enhancement for transmission over wireless access networks. *IEEE Journal on Selected Areas in Communications*, 21(2):216–228, 2003.
- [26] M. Genetzakis and V. A. Siris. A contention-aware routing metric for multi-rate multi-radio mesh networks. In *Proceedings of IEEE SECON*, June 2008.
- [27] O. Gnawali, M. Yarvis, J. Heidemann, and R. Govindan. Interaction of retransmission, black-listing, and routing metrics for reliability in sensor network routing. In *Proceedings of IEEE SECON*, Oct. 2004.

- [28] O. Gnawali, M. Yarvis, J. Heidemann, and R. Govindan. Interaction of retransmission, black-listing, and routing metrics for reliability in sensor network routing. In *Proceedings of IEEE SECON*, Oct. 2004.
- [29] X. Guo and J. Liu. Path diversified retransmission for tcp over wireless mesh networks. In *Proceedings of IEEE IWQoS*, June 2010.
- [30] D. Gupta, D. Wu, P. Mohapatra, and C. Chuah. Experimental comparison of bandwidth estimation tools for wireless mesh networks. In *Proceedings of IEEE INFOCOM*, April 2009.
- [31] M. Handley, C. Raiciu, and M. Bagnulo. Outgoing packet routing with MP-TCP, Oct. 2009. <http://tools.ietf.org/html/draft-handley-mptcp-routing-00>.
- [32] P. Hsiao, H. T. Kung, and K. Tan. Video over TCP with receiver-based delay control. In *Proceedings of ACM NOSSDAV*, 2001.
- [33] R. Jain, D.M. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared systems. *DEC Research Report TR-301*, 1984.
- [34] Hai Jiang, Weihua Zhuang, and Xuemin Shen. Cross-layer design for resource allocation in 3g wireless networks and beyond. *IEEE Communications Magazine*, 43(12):120–126, 2005.
- [35] D. Jurca and P. Frossard. Packet selection and scheduling for multipath video streaming. *IEEE Transactions on Multimedia*, 9(3):629–641, 2006.
- [36] S. Kandula, D. Katabi, S. Sinha, and A. Berger. Dynamic load balancing without packet reordering. In *Proceedings of ACM SIGCOMM*, Aug. 2007.
- [37] P. Karn and C. Partridge. Improving round-trip time estimates in reliable transport protocols. In *Proceedings of ACM SIGCOMM*, Aug. 1987.
- [38] T. Kim and M. H. Ammar. Receiver buffer requirement for video streaming over TCP. In *Proceedings of IEEE ICNP*, 2006.
- [39] P. Kyasanur and N. H. Vaidya. Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. *SIGMOBILE Mobile Computing and Communications Review*, 10(1):31–43, 2006.
- [40] Y. Lee, I. Park, and Y. Choi. Improving TCP performance in multipath packet forwarding networks. *J. Comm. and Networks*, 4(4):148–157, 2002.
- [41] K. Leung, V. Li, and D. Yang. An overview of packet reordering in transmission control protocol (TCP):problems, solutions, and challenges. *IEEE Transactions on Parallel and Distributed Systems*, 18(4):522–535, 2007.
- [42] K. C. Leung and C. Ma. Enhancing TCP performance to persistent packet reordering. *J. Comm. and Networks*, 7(3):385–393, 2005.



- [43] H. Li, Y. Cheng, C. Zhou, and W. Zhuang. Minimizing end-to-end delay: A novel routing metric for multi-radio wireless mesh networks. In *Proceedings of IEEE INFOCOM*, April 2009.
- [44] J. Li, C. Blake, and R. Morris. Capacity of ad hoc wireless networks. In *Proceedings of ACM MOBICOM*, April 2001.
- [45] W. Luo, K. Balachandran, K. Nanda, and K. K. Chang. Delay analysis of selective-repeat ARQ with applications to link adaptation in wireless packet data systems. *IEEE Transaction on Wireless Communications*, 4(3):1017–1029, 2005.
- [46] K. Manousakis and J. A. McAuley. Designing OSPF routing areas to meet diverse end-to-end performance. In *Proceedings of IEEE MILCOM*, Oct. 2007.
- [47] M. K. Marina and S. R. Das. On-demand multipath distance vector routing in ad hoc networks. In *Proceedings of IEEE ICNP*, Nov. 2001.
- [48] A. J. McAuley, K. Manousakis, and L. Kant. Flexible QoS route selection with diverse objectives and constraints. In *Proceedings of IEEE IWQoS*, June 2008.
- [49] S. Misra, G. Xue, and D. Yang. Polynomial time approximations for multi-path routing with bandwidth and delay constraints. In *Proceedings of IEEE INFOCOM*, April 2009.
- [50] B. Mukherjee and T. Brecht. Time-lined TCP for the TCP-friendly delivery of streaming media. In *Proceedings of IEEE ICNP*, Nov. 2000.
- [51] T. Nguyen and S. Cheung. Multimedia streaming using multiple TCP connections. *ACM Transactions on Multimedia Computing Communications and Applications*, 4(2):1061–1076, 2008.
- [52] T. P. Nguyen and Z. Avideh. Multiple sender distributed video streaming. *IEEE Transaction on Multimedia, Special Issue on Streaming Media*, 6(2):315–326, 2004.
- [53] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput a simple model and its empirical validation. In *ACM SIGCOMM*, Aug. 1998.
- [54] S. Pejhan, M. Schwartz, and D. Anastassiou. Error control using retransmission schemes in multicast transport protocols for real-time media. *IEEE/ACM Transactions on Networking*, 4(3):413–427, 1996.
- [55] D. S. Phatak and T. Goff. A novel mechanism for data streaming across multiple IP links for improving throughput and reliability in mobile environments. In *Proceedings of IEEE INFOCOM*, April 2002.
- [56] D. S. Phatak and T. Goff. A novel mechanism for data streaming across multiple IP links for improving throughput and reliability in mobile environments. In *Proceedings of IEEE INFOCOM*, April 2002.

- [57] R. Puri, K. Lee, K. Ramchandran, and V. Bharghavan. An integrated source transcoding and congestion control paradigm for video streaming in the internet. *IEEE Transactions on Multimedia*, 3(1):18–32, 2001.
- [58] B. Radunovic, C. Gkantsidis, D. Gunawardena, and P. Key. Horizon: Balancing TCP over multiple paths in wireless mesh network. In *Proceedings of ACM MOBICOMM*, Sep. 2008.
- [59] B. Radunovic, C. Gkantsidis, P. Key, and P. Rodriguez. An optimization framework for opportunistic multipath routing in wireless mesh networks. In *Proceedings of IEEE INFOCOM*, April 2008.
- [60] D. Cheriton S. Liang. TCP-RTM: Using TCP for real time applications. In *Proceedings of IEEE ICNP*, 2002.
- [61] P. Sarolahti. Multipath interface in the socket API, Nov. 2009. <http://tools.ietf.org/html/draft-sarolahti-mptcp-af-multipath-00>.
- [62] C. Sarr, C. Chaudet, G. Chelius, and I. G. Lassous. Bandwidth estimation for IEEE 802.11-based ad hoc networks. *IEEE Transactions on Mobile Computing*, 7(10):1228–1241, 2008.
- [63] M. Scharf and A. Ford. MPTCP application interface considerations, Oct. 2009. <http://tools.ietf.org/html/draft-scharf-mptcp-api-00>.
- [64] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod. Cross-layer design of ad hoc networks for real-time video streaming. *IEEE Wireless Communications*, 12(1):59–65, 2005.
- [65] K. Sripanidkulchai, B. Maggs, and H. Zhang. An analysis of live streaming workloads on the internet. In *Proceedings of ACM SIGCOMM Conference on Internet Measurement*, 2004.
- [66] W. Stevens. TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms, Jan. 1997. <http://www.ietf.org/rfc/rfc2001.txt>.
- [67] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao. Minimum interference channel assignment in multiradio wireless mesh networks. *IEEE Transactions on Mobile Computing*, 7(12):1459–1473, 2008.
- [68] W. Tarn and Y. Tseng. Joint multi-channel link layer and multi-path routing design for wireless mesh networks. In *Proceedings of IEEE INFOCOM*, May 2007.
- [69] J. W. Tsai and T. Moors. Minimum interference multipath routing using multiple gateways in wireless mesh networks. In *Proceedings of IEEE MASS*, Sep. 2008.
- [70] J. van der Merwe, S. Sen, and C. Kalmanek. Streaming video traffic: Characterization and network impact. In *Proceedings of the Seventh International Web Content Caching and Distribution Workshop*, 2002.

- [71] J. van der Merwe, S. Sen, and C. Kalmanek. Streaming video traffic: Characterization and network impact. In *Proceedings of the Seventh International Web Content Caching and Distribution Workshop*, Aug. 2002.
- [72] B. Wang, J. Kurose, P. Shenoy, and D. Towsley. Multimedia streaming via TCP: An analytic performance study. *ACM Transactions on Multimedia Computing Communications and Applications*, 4(2), 2008.
- [73] B. Wang, J. Kurose, P. Shenoy, and D. Towsley. Multipath live streaming via TCP: Scheme, performance and benefits. *ACM Transactions on Multimedia Computing Communications and Applications*, 5(3), 2009.
- [74] Z. Wang and J. Crowcroft. Finding disjoint paths in networks. *ACM SIGCOMM Computer Communication Review*, 21(4):43–51, 1991.
- [75] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 14(7):1228–1234, 1996.
- [76] W. Wei and A. Zakhori. Path selection for multi-path streaming in wireless ad hoc networks. In *Proceedings of IEEE ICIP*, Oct. 2006.
- [77] D. Wu and P. Mohapatra. From theory to practice: Evaluating static channel assignments on a wireless mesh network. In *Proceedings of IEEE INFOCOM*, March 2010.
- [78] W. Wu, P. Demar, and M. Crawford. Sorting reordered packets with interrupt coalescing. *Computer Networks*, 53(15):2646–2662, 2009.
- [79] Z. Wu, S. Ganu, and D. Raychaudhuri. IRMA: Integrated routing and MAC scheduling in multi-hop wireless mesh networks. In *Proceedings of IEEE WiMesh*, Sep. 2006.
- [80] Jiangchuan Liu Xiaoyuan Guo and Shiguo Lian. Real-time video streaming over multipath in multi-hop wireless networks. *ACM Multimedia Systems*, 17(4):287–297, 2010.
- [81] D. Xu, Y. Chen, Y. Xiong, C. Qiao, and X. He. On finding disjoint paths in single and dual link cost networks. In *Proceedings of IEEE INFOCOM*, March 2004.
- [82] Kai Xu, Ye Tian, and N. Ansari. TCP-Jersey for wireless IP communications. *IEEE Journal on Selected Areas in Communications*, 22(4):747–756, 2004.
- [83] Y. Yang, J. Wang, and R. Kravets. Designing routing metrics for mesh networks. In *Proceedings of IEEE WiMesh*, Sep. 2005.
- [84] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi. Effects of multipath routing on TCP performance in ad hoc networks. In *Proceedings of ICC*, April 2003.
- [85] K. Zeng, Z. Yang, and W. Lou. Opportunistic routing in multi-radio multi-channel multi-hop wireless networks. In *Proceedings of IEEE INFOCOM*, March 2010.

- [86] M. Zhang, B. Karp, S. Floyd, and L. Peterson. RR-TCP: A reordering-robust TCP with DSACK. In *Proceedings of IEEE ICNP*, Nov. 2003.
- [87] W. Zhang, J. Tang, C. Wang, and S. Soysa. Reliable adaptive multipath provisioning with bandwidth and differential delay constraints. In *Proceedings of IEEE INFOCOM*, March 2010.
- [88] B. Zhou, C. Fu, D. Chiu, C. Lau, and L. Ngoh. A simple throughput model for TCP Veno. In *Proceedings of IEEE ICC*, 2006.
- [89] C. Zhu and M. S. Corson. QoS routing for mobile ad hoc networks. In *Proceedings of IEEE INFOCOM*, June 2002.