2011-11-14

Modeling and Designing Fair Rate Control for Wireless Mesh Networks with Partial Interference

Lei Wang
Brigham Young University - Provo

Modeling and Designing Fair Rate Control for Wireless Mesh Networks With Partial Interference

Lei Wang

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Daniel M.A. Zappala, Chair
Sean C. Warnick
Kent E. Seamons
Charles D. Knutson
Mark J. Clement

Department of Computer Science

Brigham Young University

December 2011

ABSTRACT

Modeling and Designing Fair Rate Control for Wireless Mesh Networks
With Partial Interference

Lei Wang
Department of Computer Science, BYU
Doctor of Philosophy

Internet rate control protocols, such as TCP, encounter severe performance problems in wireless mesh networks. Because wireless networks use shared communication channels, contention and interference can significantly degrade flow throughput and fairness. Existing research takes either an engineering-based or optimization-based approach to solve the performance problems. The engineering-based approach usually solves a specific observed problem, but does not necessarily optimize the overall performance. The optimization-based approach mathematically models the network to find the optimal resource allocation among competing flows. The model can lead to a distributed rate control algorithm with performance guarantees, but relatively little work has been done to verify that the algorithm leads to good performance in real networks.

This dissertation develops a more accurate network optimization model, implements the derived distributed rate control algorithm in a mesh testbed, and discusses observations in the extensive experiments. We first synthesize models used for optimizing fair rate control for wireless mesh networks, and discuss their tradeoffs. We then propose a partial interference model which uses more accurate objective functions and constraints as compared to the binary interference model. Numerical results show that the partial interference model outperforms the binary interference model in all scenarios tested, and the results also suggest that partial interference should be modeled separately from contention. Our experimental results confirm the prevalence of partial interference in our mesh testbed, and show that the partial interference model results in significantly improved performance in a typical interference topology. We also observe a significant deviation between theory and practice, whereby, the assumption of a linear relationship between interfering links breaks in our experiments. We discuss several directions to further investigate this issue.

Keywords: Wireless mesh networks, fair rate control

ACKNOWLEDGMENTS

To my wife, Qiuyi, I wouldn't have been able to get here without your great love and support to accompany me through all the long nights and frustrations. Life is hard, but we've been through all the difficulties together, and my life becomes a lot more colorful and meaningful simply because of you.

To my advisor Dr. Daniel Zappala and Dr. Sean Warnick, thank you for all the great support and suggestions. Most of all, thank you for teaching me to always keep faith in myself. What I've learned under your instructions goes far beyond the school to the rest of my life.

Last but not least, I would like to thank everyone who offered the great help when it is needed most. Words are not enough to express my gratitude, but THANK YOU!

Table of Contents

List of Figures vii

List of Tables ix

1 Introduction 1

2 Related Work 5

2.1 Engineering-Based Approaches 5

2.1.1 End-To-End Approaches 6

2.1.2 Hop-By-Hop Approaches 7

2.1.3 Cross-Layer Approaches 9

2.2 Optimization-Based Approaches 11

3 Challenges in Multi-Hop Wireless Communications 15

4 Modeling Wireless Mesh Networks 19

4.1 Resource Constraints 19

4.2 Objective Functions 22

4.3 Basic Units in Resource Allocation 25

5 Partial Interference Model 33

5.1 Resource Constraints 33

5.2 Objective Functions 35

5.3 Link-Based Problem Formulation 36

5.4 Non-Convexity in the Flow-Based Formulation 38
5.5 Distributed Algorithm 39

6 Numerical Results **43**
6.1 Performance metric 43
6.2 Results 44

7 Protocol Implementation **49**
7.1 Implementation Goals 49
7.2 Wireless Mesh Testbed 50
7.3 Network Interference and Contention Map Measurement 51
7.4 Maximal Clique Enumeration 55
7.5 User-space Development Toolkit 57
7.6 Fair Rate Control 59

8 Experimental Results **65**
8.1 Experiment Configurations 66
8.2 Network Interference and Contention Map Measurement 68
8.3 Partial Interference Scenario 72
8.3.1 Parital Interference Model 72
8.3.2 Interference as Contention Model 76
8.3.3 Interference Ignored Model 78
8.3.4 Sum Utility Comparison 78
8.4 Pure Contention Scenario 80

9 Conclusions **83**

References **85**

List of Figures

1.1 Wireless mesh network 1

3.1 Three important ranges in wireless transmissions 16

3.2 Problems in wireless transmissions 17

4.1 A network topology graph. 21

4.2 A contention graph for the sample network topology (binary interference model) . 21

4.3 Example network topology 1 29

4.4 Example network topology 2 31

5.1 A contention graph for the sample network topology (partial interference model) . 34

6.1 Topologies used for numerical results. 45

6.2 Numerical results for I interferers on one link. 46

6.3 Numerical results for N contenders with one interferer. 47

6.4 Numerical results for I interferers and N contenders, with $a = 0.4$. 48

7.1 Wireless mesh testbed first floor nodes 51

7.2 Fair rate control implementation architecture 59

7.3 Fair rate control message format 61

7.4 Fair rate control messages 62

8.1 Partial interference topology 69

8.2 Interference factor measurement 70

8.3 Interference factor measurement (transmssion power 9) 71

8.4 Broadcast vs. unicast 71
8.5 Flow throughput (PI) 73
8.6 Outgoing packet arrival intervals at node 17 74
8.7 Outgoing packet arrival intervals at node 25 74
8.8 Non-linear relationship between interferer and interferee links 76
8.9 Fair rate convergence with IC model 77
8.10 Flow throughput (IC) 77
8.11 Flow throughput (II) 78
8.12 Sum utility 79
8.13 Sum utility performance value as a function of interferer rate 80
8.14 Hidden terminal topology 81
8.15 Flow throughput and fair rate convergence (PI) 81
8.16 Sum utility 82

List of Tables

4.1 Notations used in the formulation 25

Chapter 1

Introduction

A wireless mesh network consists of a mesh of stationary and mobile devices, connecting to each other through wireless links. Figure 1.1 illustrates a typical mesh network, in which a set of stationary wireless routers maintain a backhaul network, providing basic wireless coverage. They usually have unlimited power supply and powerful CPUs, and work as traffic forwarders in the network. Users access the network using mobile devices such as laptop computers and smart phones, and they may move around the mesh network, connecting through wireless routers. A mesh network often includes one or more gateways to the Internet, providing extended access to Internet services. As compared to a wireless access point using a single wireless hop, communication paths in a typical mesh network span *multiple* hops before reaching the wired network.

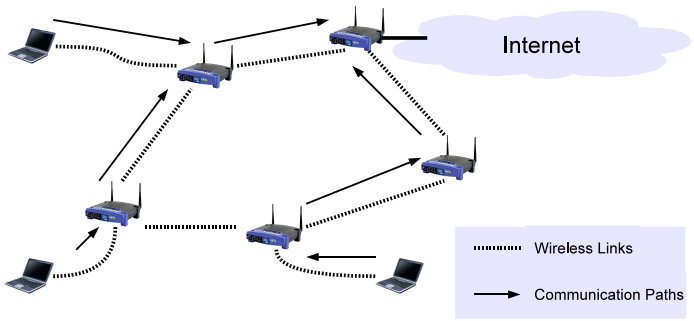


Figure 1.1: Wireless mesh network

Wireless mesh networks are becoming increasingly common, both as standalone networks and as extensions to the Internet services. Usually smaller than desktop computers, wireless routers can be easily deployed wherever power is available without worrying about the expensive infras-

tructure needed to provide a wired Internet connection at every place where access is needed. Intelligent network protocols make mesh networks self-healing in that a host is able to choose an alternative communication path in situations when an intermediate node in the original path becomes unavailable or encounters severe problems in link quality. Wireless mesh networks have traditionally been used in situations where an extensive Internet coverage is not affordable, such as in rural or underprivileged areas, or when an infrastructure network coverage is not available at the moment, such as in disaster relief situations. Today, the most common use is providing video surveillance in cities where it is prohibitively expensive to wire the whole city. Representative projects include the One Laptop Per Child (OLPC) project at MIT [1], the Technology for All (TFA) project at Rice University [2], and the mesh network deployed in the San Francisco Bay area [3].

It is well-known that Internet rate control protocols experience severely degraded throughput and fairness in wireless mesh networks. This is because link conditions in wireless mesh networks are significantly different from those in the Internet. Internet rate control protocols were originally designed to work for wired Internet connections, through which disjoint flows won't affect each other. In a wireless mesh network, however, transmissions are broadcast in nature, and those in close vicinity cause contention and interference, which often results in poor throughput and unfairness between competing flows [4, 5]. Performance evaluation suggests that contention and interference are major reasons for packet loss in wireless mesh networks, and extra rate control is necessary to minimize the undesirable contention and interference in the network.

To cope with these problems, this dissertation develops a fair rate protocol that 1) works transparently to existing Internet rate control protocols; 2) maximizes the overall link utility in the wireless mesh network; and 3) achieves proportional fairness between competing wireless links. We first design a theoretic model for a wireless mesh network that incorporates partial interference, which is usually overlooked in existing models. Using the partial interference model, we then derive a distributed rate control algorithm with performance guarantees, and present numerical results to show the benefits of the algorithm as compared to more conservative models. We

also implement the rate control algorithms in a wireless mesh testbed, and evaluate performance through extensive experiments.

In the theoretical modeling part of this dissertation, we seek to answer a fundamental question in modeling wireless mesh networks as applied to rate control: is it important to model partial interference separately from contention? We discuss limitations in the traditional binary interference model, propose our partial interference model, and formulate the optimization problem. We discuss tradeoffs in designing practical rate control algorithms using the proposed model and use numerical results to illustrate conditions under which the effects of partial interference cannot be ignored and should be modeled accurately.

Our experimental results on a mesh testbed show that partial interference is prevalent in the mesh network, and treating interference as contention usually leads to over-conservative resource allocations. Thus our partial interference model does improve overall throughput received by applications. The interference model uses a network interference map to calculate the level of impact that an interferer link inflicts on an interferee link. However, we observe that measuring the interference map using broadcast may lead to inaccurate results. A unicast approach is preferable in terms of measurement accuracy. We also find a non-linear relationship between the interfering links. This contradicts the seminal work in the field that established a linear relationship based on measurements of a mesh network [6]. The non-linear relationship leads to a significant gap between theory and practice, demonstrating a need for further research in the area.

Chapter 2

Related Work

The research community has undergone intensive work in solving the performance problems that Internet rate control protocols encounter in wireless mesh networks. We broadly categorize existing solutions into two groups: engineering-based approaches, and optimization-based approaches. Under the engineering-based category, proposals could be further categorized as solutions that work on transport-layer flow ends, that operate at intermediate hops of flows, or that require close collaboration across multiple layers of the network.

2.1 Engineering-Based Approaches

In this category, research teams identify performance problems through experimental observations. Ad hoc solutions are then developed, implemented, and evaluated in practical network environments. The advantage of such research is that it is intimately connected to practice, leaving no gap between ideas and implementation. On the other hand, such solutions may be ad hoc and very narrow, only applying to the specific situation observed in practice without any theory for adapting them to new situations or providing performance guarantees. Research groups approach the problems from different perspectives, including better rate control algorithms on flow ends, more accurate and prompt reaction to link dynamics in the middle of flows, and better collaborations across protocol stack layers.

2.1.1 End-To-End Approaches

End-to-end congestion control schemes improve TCP performance by designing better algorithms at the source and destination of a flow. Protocols in this category inherit the original design of TCP from the Internet, where intelligence is pushed to the edge of the network for scalability considerations [7].

TCP with Adaptive Pacing improves TCP performance by adopting a rate-based congestion control algorithm [8]. TCP-AP considers both inter-flow and intra-flow contentions in estimating the packet sending rate. For inter-flow contentions, TCP-AP uses the coefficient of variation of recently measured Round Trip Times (RTT), and for intra-flow contentions it uses the measured 4-hop propagation delay. The authors in this work also recognize the importance of reducing ACK-incurred overhead to protocol performance, and adopt delayed ACKs as proposed in a previous work [9]. With delayed ACKs, a transport protocol can combine up to four ACK packets into one segment. The authors justify the protocol designs and verify protocol performance through a simulation study. Unfortunately, the practical performance of TCP-AP largely depends on the network topology and traffic patterns.

TCP with Fractional Window Increment (FeW) limits TCP's aggressiveness in congestion control to achieve a better performance and interaction with on-demand routing protocols [10]. This research work identifies that network overload is the primary reason for network performance degradation, and the bad interactions between on-demand routing protocols and TCP make the situation even worse. The size of the congestion window in TCP determines the number of outstanding packets in the network; this work avoids network overload by using a fractional window increment scheme. Rather than additively increasing the congestion window after the successful transport of a window of packets, TCP-FeW fractionally increases the size of the congestion window. This work demonstrates that an appropriate choice of the fraction value in congestion window increment can significantly improve TCP performance, but leaves the explicit algorithm to adaptively determine the right factional value unexplored. The enhancement in TCP performance in this work only comes from a simulation study using a static network topology.

TCP-DOOR improves TCP performance in situations where temporary link failures and route changes happen frequently [11]. TCP interprets packet loss as a sign of congestion. This assumption works fine in the Internet, but breaks in the world of multi-hop wireless networks, where packet loss could also be a result of route changes and interference. TCP-DOOR infers route changes from out-of-ordered (OOO) packet delivery events at both ends of a flow: the source node detecting OOO ACK packets and the destination node detecting OOO data packets. The destination node needs to notify the source node about the OOO events. The source node responds to OOO events by temporarily disabling congestion control and recovering instantly to the state before the congestion avoidance action. While proposing a novel point of view to improve TCP performance, TCP-DOOR also bears a few limitations in its design. This work only provides performance reports from single-flow scenarios, which are rare cases in practice. The effectiveness of TCP-DOOR in situations where packet loss is not caused by route changes is unclear.

TCP-ATL seeks a unified solution to reliable packet transport over heterogeneous wireless media [12]. The protocol uses an Exponential Weighted Moving Average (EWMA) in calculating the estimated RTT and deviation in sampled RTTs. The parameters α and β determine the responsiveness to variations in measured RTT values. TCP-ATL adaptively adjusts the value of these two parameters to cope with the dramatically different characteristics in wireless media, which are captured by the packet loss rate and wireless link delay observed at the MAC layer. TCP-ATL emphasizes issues that come from heterogeneity in wireless media, while this work seeks enhancement in transport performance in IEEE 802.11-based wireless mesh networks.

2.1.2 Hop-By-Hop Approaches

Hop-by-hop approaches perform congestion control at intermediate nodes along a communication path. In contrast to end-to-end schemes, hop-by-hop congestion control pushes intelligence into the network. The relatively small scale of wireless mesh networks justifies the viability of such designs. Although hop-by-hop designs originated on wired networks, the application of hop-by-hop design in multi-hop wireless networks aims to overcome unique challenges in wireless transmis-

sions.

Hop-by-hop congestion control is particularly intriguing in sensor networks, where maximizing the battery life of sensors is a primary challenge. With hop-by-hop congestion control, nodes avoid congestion and recover lost packets at intermediate nodes, conserving more power than with end-to-end congestion control. Unfortunately, protocols tailored for sensor networks usually lack the generality needed to be applied to other types of multi-hop wireless networks, such as mesh and ad hoc networks. Most of these protocols take advantage of the unique attributes in sensor networks, which include a many-to-one traffic pattern, homogeneous packet sizes and transmission rates, and usually no mobility. A typical multi-hop wireless mesh network presents significantly more dynamics in the number of flows, transmission rates, and network topology.

A good example of hop-by-hop congestion control in sensor networks is Fusion [13]. Each sensor monitors its queue length and sets a bit in its outgoing packets when the queue grows too large. An upstream sensor overhears this information, and stops transmitting to this sensor until it overhears a packet with the bit cleared. In this way local congestion information is carried towards the source node via backpressure. Fusion also adopts a rate limiting scheme to alleviate the serious unfairness toward sources that have to traverse a larger number of wireless hops. Each sensor listens to the traffic its parent forwards to estimate the total number of unique sources N that route through the parent and uses a token bucket scheme to limit the sending rate to $1/N$ of the total rate. Fusion also adopts a prioritized MAC layer that gives a backlogged sensor priority over non-backlogged sensors for access to the shared wireless medium, thus avoiding buffer drops. Similar work in sensor networks includes CODA [14], ARC [15] and CCF [16].

In a more recent work, Scofield et al. propose HxH, a hop-by-hop transport protocol for wireless mesh networks [17, 18]. HxH improves throughput by efficient designs in hop-wise congestion control and end-to-end reliability. At each hop, a node overhears ongoing transmissions at the downstream node and estimates the size of available buffers at the next hop. To avoid congestion nodes only transmit when there are available buffers at the downstream node. The node that is one hop away from the destination node piggybacks in its transmissions the sequence number

of the packet last received by the destination. The node one hop further away from the destination can overhear this and continue the process. In this way, the end-to-end reliability information is passively relayed to the source node, conserving scarce wireless resources for data transmissions. The credit-based congestion control in HxH, however, may still send at a high rate, and it may generate bursty traffic and thus exaggerate contention in network.

2.1.3 Cross-Layer Approaches

Protocol designers also seek approaches to improve TCP performance through a joint effort across multiple layers in the protocol stack, in particular from network and link layers. Typical work in this category maintains the end-to-end semantics of TCP. Intermediate nodes feedback link conditions in the communication path back to the source node to help TCP react more accurately.

It is well-recognized that TCP fails to distinguish the difference between a link failure and congestion and reacts erroneously to link dynamics in the network. Extensive research work has proposed solutions to this problem. A good example is Explicit Link Failure Notification (ELFN), in which a network layer protocol notifies TCP when a route has failed in mobile ad hoc networks [19]. TCP freezes the retransmission timer and enters a "stand-by" mode, giving the routing protocol time to repair the route failure. In order to determine when the route has been restored, TCP sends periodic probe packets to see if a route has been established. If an acknowledgement is received, it then leaves the "stand-by" mode, restores its retransmission timers, and continues as normal. In this way, TCP effectively distinguishes mobility loss from congestion-incurred loss, and avoids unnecessary reductions in its sending rate.

As an effort to further reduce the impact of mobility-incurred packet loss on TCP performance, Yu proposes to use intermediate nodes to improve end-to-end performance by two mechanisms: Early Packet Loss Notification (EPLN) and Best-Effort ACK Delivery (BEAD) [20]. This work extensively exploits cached routes at the network layer to help TCP cope with route changes in mobile ad hoc networks. Upon a route failure, an intermediate node salvages packets by sending them on an alternative, cached route. If packet salvation fails, EPLN notifies TCP about lost

packets during the route failure. TCP disables its retransmission timer to avoid an unnecessary decrease in its sending rate. BEAD attempts to retransmit ACKs at either intermediate nodes or TCP receivers, alleviating the impact of lost ACK messages on flow throughput.

Ad-hoc Transport Protocol (ATP) is a rate-based transport protocol tailored for ad hoc networks [21]. In this work Sundaresan et al. argue that several design elements in TCP, such as window-based transmissions and loss-based congestion indication, are fundamentally inappropriate for the unique challenges in ad-hoc networks. In order to overcome these observed disadvantages, ATP measures queuing delay and transmission delay at each intermediate nodes; the measurements are piggybacked on data packets to the destination. The destination relays the sum of the measurements to the source to control the sending rate. To reduce communication overhead, the destination also sends ACKs at epochs instead of for every packet. Even though ATP is designed to replace TCP in ad hoc networks, recent research demonstrates that ATP is unable to maintain stable transmission rates and usually chooses rates that are much lower than necessary when the network topology becomes highly dynamic [17].

2.2 Optimization-Based Approaches

In order to overcome the limitations in the "ad-hoc" design pattern of the engineering-based approaches, researchers have appealed to convex optimization theory to systematically formulate rate control problems and derive solutions that yield performance guarantees. Proposals in this category typically aim to maximize the overall utility in the network, subject to constraints imposed by link capacities. Depending on the definition of utility used in the model [22, 23], the resulting solutions can flexibly achieve different senses of fairness, such as max-min fairness and proportional fairness. Distributed algorithms capable of computing these solutions are then derived via various decomposition methods [24].

The seminal work from Kelly applies theory from convex optimization to solve the problem of optimal resource allocation in a communication network [25]. This work models the optimal resource allocation as a problem that maximizes the sum of flow utilities, which are functions of

rate assigned to each flow, with constraints imposed by link capacity and the traffic flow pattern. Following a convention in convex optimization, this paper interprets slack variables as the price per unit flow that the network charges a user. This work then models the problem from a user's point of view as a maximizer of the received utility, and from the network's point of view as a maximizer of the received revenue from all users. A theoretical proof suggests that there exists a equilibrium satisfying both sides. This work assumes that the utility function is differentiable and strictly concave so that the optimization problem attains the favorable convex attribute. Standard techniques can be used to derive distributed algorithms for convex problems.

In the same paper, Kelly proposes the concept of proportional fairness in resource allocation. The traditional concept of max-min fairness gives every flow with equal demand an equal share; proportional fairness emphasizes aggregated network throughput, and allows some degree of "unfairness" in the throughput of some flows as long as this sacrifice can achieve a greater increment in the aggregated network utility. This paper, however, predates many recent developments with wireless networks and thus does not take into account the shared nature in wireless transmissions. As critiqued in a subsequent work, a direct application of the derived congestion control protocol in the wireless world leads to an unstable equilibrium point at the desired fair solution [26].

Inspired by Kelly's work, Yi et al. take a similar approach and develop a hop-by-hop congestion control protocol for multi-hop wireless networks [27]. Each node collects the sum of MAC time utilization by all traversing flows, both incoming and outgoing, and calculates a local congestion price as the difference between this sum and a specified utilization threshold, which is determined by the efficiency of the MAC protocol in use. Each node adds its current congestion price to the price it received from a downstream node, and passes this partial sum toward the upstream node. The source node ultimately receives the sum of all price information from the corresponding nodes on its path, and uses this price to control its rate. In addition to the rate adjustment at the source, each intermediate node also controls its sending rate based on the received partial sum and achieves a more responsive reaction to changes in link conditions.

The proposed algorithm assumes that each wireless hop operates on a separate frequency, and the congestion price feedback does not experience any delay and loss. In the performance evaluation, the desired MAC utilization threshold is manually set without reporting the method to derive this value in practice.

In another representative work, Chen et al. propose a joint congestion control and media access control model for ad hoc wireless networks [28]. Recognizing the shared nature of the wireless medium, the authors apply the concept of a maximal clique in graph theory to model the contention relationship among contending transmissions [1]. Each intermediate node collects flow rate information from every other node within the same clique, and calculates a congestion price as a function of the normalized sum rate. A source node adjusts its sending rate according to the cumulative congestion price that is periodically fed back from downstream nodes in the communication path. At intermediate hops, each flow uses the normalized flow rate as a persistence probability to contend for the channel. This paper, along with many other models that use a contention graph, ignores the impact of partial interference, which is prevalent in wireless mesh networks and may significantly impact the performance of rate control as shown in more recent publications [6, 29, 30].

This work also points out that the the link capacity constraint is only a necessary condition to realizable rate allocations. The link capacity constraint suggests that the sum of the transmission rates in a contention domain should not exceed the link capacity, because they have to share the common channel. A rate allocation that satisfies the constraint is not realizable if the contention graph contains a hole with an odd number of vertices. If a graph is perfect, the constraint then becomes the sufficient condition to realizable rate allocations. The authors note that identifying whether a graph is perfect requires global topology information of the network, which is impractical for designing distributed algorithms. The capacity of a clique is decreased by some fraction in the proposed optimization problem in order to guarantee the scheduling feasibility of a rate allocation. Our dissertation focuses on proposing a more accurate model of wireless mesh networks

[1] We will further discuss the concept of maximal clique and contention graph in the following chapter.

and evaluating the performance of the proposed rate control in practice. We assume that the contention graph in our network is perfect and decrease the clique capacity in our implementation when necessary.

Taking a similar convex optimization approach, different decompositions have lead to numerous models during the past few years. Representative work includes cross-layer congestion control, routing and scheduling design [31], jointly optimal congestion control and routing [32], and joint congestion control and physical layer power control [33].

The strengths of these approaches are their systematic methodology and provable performance guarantees. These guarantees, however, only apply when the conditions of the formulation are satisfied, or when the resulting distributed algorithm is clearly implementable. Unfortunately, the simplifications needed to make such formulations tractable are often either over-simplified from practical conditions or too strong to admit practical implementations. Moreover, while much of the published work in this category focuses on correctness proofs of the resulting algorithms, the question of whether practical implementations exist that satisfy the conditions for these proofs is not adequately addressed; convincing experimental results demonstrating the practicality of the theory are absent. This dissertation contributes valuable insights with experimental results from practical wireless mesh networks.

As a preliminary effort to overcome the above pitfalls, the work presented in [34] addresses those practical challenges and reports experimental results from a real network. As a workaround to enumerating maximal cliques in a contention graph, which is known to be NP-hard, the proposed protocol simply assigns nodes within two hops to the same clique. This is an overly conservative design. In order to guarantee that every node in a clique attains the same view of the clique size and membership, link declaration messages are sent over separate channels and are forwarded over three hops. As an effort to reduce bandwidth consumption, flow rates are aggregated in batches and exchanged less frequently. Despite the efforts in the efficiency considerations, the protocol may take over 20 seconds to converge in simple static topologies. Unfortunately, this convergence rate is un-acceptable for practical networks.

Chapter 3

Challenges in Multi-Hop Wireless Communications

We first look at the complications that occur in wireless transmissions. In this section we use a simple example to demonstrate why reliable wireless transmissions are problematic in wireless mesh networks.

We first explain three import ranges in wireless transmissions: *transmission range*, *carrier sense range*, and *interference range*. As depicted in Figure 3.1, the transmission range identifies the distance within which nodes can successfully receive and decode frames from a transmitting node if there is no interference from other sources. Many research works over-simplify the transmission range as a circular area centered at the transmitting node. In practice, the transmission range varies in different directions, typically affected by physical obstacles and interference, and generally takes an irregular shape. The carrier sense range is the range within which a transmitting node triggers carrier sense detection. This range is usually determined by the antenna sensitivity (physical capability) and a human-set threshold, above which the carrier is considered busy. The IEEE 802.11 MAC regulates that a node can only transmit when it senses a clear carrier (wireless channel). The interference range is the range within which unrelated signals become strong enough such that a receiving node cannot distinguish its desired signal from noise, and thus suffers frame loss. The carrier sense range is usually larger than transmission range [35, 36]. The relationship between the transmission range and the interference range is determined by the transmitting power, the distance and physical condition between the communicating nodes, and the antenna sensitivity at receiving nodes. In many situations the interference range is larger than transmission range, and the power level needed for interrupting a transmission is much smaller than that of successfully

delivering a packet [35].

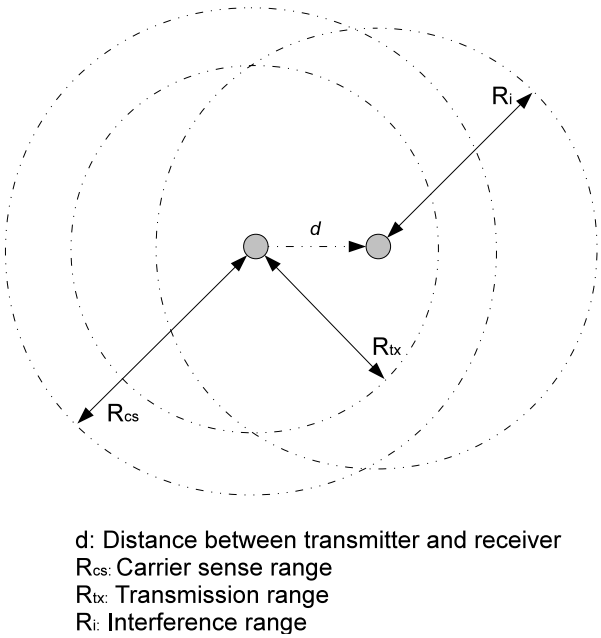


Figure 3.1: Three important ranges in wireless transmissions

The IEEE 802.11 MAC resolves collisions in wireless transmissions via a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. Each node senses the carrier before any transmission attempts. A node only transmits if the carrier remains clear over a period of time. The IEEE 802.11 MAC enforces randomness in the duration that nodes must wait before each transmission to avoid collisions between concurrent carrier sensing. The optional Request To Send, Clear To Send (RTS/CTS) message exchange process is designed to further prevent collisions incurred by hidden terminals [35]. After sensing a clear channel, a transmitting node sends an RTS message to the receiving node, declaring its intention to transmit a frame. With CTS, a receiving node grants the sending node the right to transmit, and also implicitly notifies neighboring nodes of the coming data transmission.

Figure 3.2 demonstrates several cases where collision, contention, and interference affect frame transmissions in multi-hop wireless mesh networks. Assume each node has a transmission range of 150m and an interference and carrier sensing distance of 250m. Nodes are 100m away

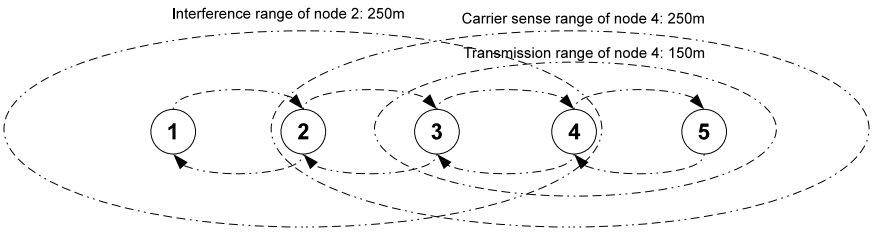


Figure 3.2: Problems in wireless transmissions

from each other.

1. One-hop case: suppose both node 1 and node 2 want to transmit frames at the same moment. Their transmissions collide if both nodes transmit without any regulation. If they use the CSMA/CA mechanism in IEEE 802.11 MAC to resolve collisions they share the channel, and each of them receives approximately half of the bandwidth.

2. Two-hop case: suppose node 1 intends to send packets to node 3 via node 2. With only one radio, node 2 cannot receive and send frames simultaneously. The transmission from node 1 to 2 and from node 2 to 3 have to contend for the shared wireless medium, and their achievable bandwidth is halved.

3. Three-hop case: suppose node 1 wants to send packets to node 4 via node 2 and 3. Concurrent transmission from node 3 to node 4 may completely or partially collide with those from node 1 to node 2 at node 2. Some protocols constrain node 3 from transmitting concurrently with node 1 to avoid collisions, and nodes can only achieve one third of the total bandwidth of the wireless channel.

4. Four-hop case: hidden terminal effect. Assume node 1 sends packets to node 5 along a path through node 2, 3, and 4. Node 4 becomes a hidden terminal to transmissions from node 1 to 2 in the sense that node 1 does not realize the presence of transmissions from node 4 since it is out of the sensing range of node 1. As a consequence, transmissions originating at node

1 may be corrupted at node 2 by transmissions from node 4, which is within the interference range of node 2. Nodes can only attain one fourth of the bandwidth of the wireless channel.

In practice, network environments become far more complicated than scenarios in Figure 3.2. Rather than a simple chain topology, nodes are more likely to be deployed in a grid or mesh form, which makes transmissions more prone to contention and interference. Because they are exposed in the open air, wireless transmissions are also vulnerable to external inference such as signals from a cordless phone or other running wireless networks, or temporal interruptions from moving obstacles such as humans and vehicles. Such factors lead to a highly dynamic and unpredictable environment for wireless transmissions, and significant challenges to protocol designs for wireless mesh networks.

Chapter 4

Modeling Wireless Mesh Networks

In this section we synthesize models used for optimizing fair rate control for mesh networks. Among research using the optimization-based approach, many models have been proposed to characterize fair resource allocation for mesh networks. However, there is no work discussing when one model is preferable over the other. This section summaries major components in modeling wireless mesh networks and discusses their tradeoffs. For the purpose of designing rate control algorithms, a model needs to determine 1) how competing transmissions share the common wireless channel; 2) the objective of an optimization problem; and 3) how to choose the basic units that participate in the rate allocation.

4.1 Resource Constraints

Resource constraints impose boundaries to transmission rates so that the rate allocation derived using the model is realizable in practice. Given a single transmission, the feasible rate should be non-negative and no greater than the link capacity, which is usually normalized to 1 in the modeling. As typical wireless mesh networks consist of a group of nodes, resource constraints are usually characterized by a set of neighboring transmissions, which exclusively compete for the shared wireless channel. Intuitively, the sum of the transmission rates should not exceed the link capacity.

Two transmissions could compete with each other on either the sending or the receiving side. Transmissions that compete on the sending side are considered to be contending with each other, while those that compete on the receiving side are treated as interfering with each other. In

the IEEE 802.11 MAC, transmissions with senders within carrier sense range exclusively contend for the shared channel. As a result, the sum of their sending rates should not exceed 1. On the receiving side, two transmissions interfere with each other if frames from one corrupt the reception of another. Channel-based models and graph-based models are two major approaches in the literature to characterize the behaviors of concurrent transmissions for this purpose [37].

Channel-based models typically use the signal-to-interference-plus-noise ratio (SINR) to determine whether a transmission will succeed or not [38]. A node can successfully decode a received frame if the $\frac{S}{I+N}$ at the receiving moment is above some certain threshold, where S, I, N stand for the power level of the signal, interference, and background noise respectively. However, channel-based models are considered to be intractable by protocol designers because of the mathematical complications [39]. Moreover, most commodity wireless cards do not provide statistics about SINR, making it difficult to collect the desired information in practice.

In this dissertation, we consider graph-based models, and investigate methods to improve their accuracy so that protocol designers can derive practical rate control algorithms for wireless mesh networks. Existing graph-based models conservatively characterize interference as a binary effect [28, 37]. Under this model, an interfering transmission is assumed to corrupt all of the frames received at a remote node, while non-interfering nodes have no effect. Binary interference is represented in a contention graph by simply treating it as contention, that is, if one link interferes with another, neither may send at the same time.

To illustrate how the binary interference model derives resource constraints, it is useful to consider an example. Fig. 4.1 shows a sample network topology, denoting active transmissions, transmission ranges, carrier sense ranges, and interference ranges. A contention graph transforms this representation of a network into a new graph that represents the contention and interference constraints [40, 28]. Fig. 4.2 shows a contention graph for Fig. 4.1. Vertices in the contention graph correspond to wireless links, and an edge between two links indicates that the links cannot be active at the same time, due to contention or interference.

Once a contention graph is created, resource constraints can be determined using maximal

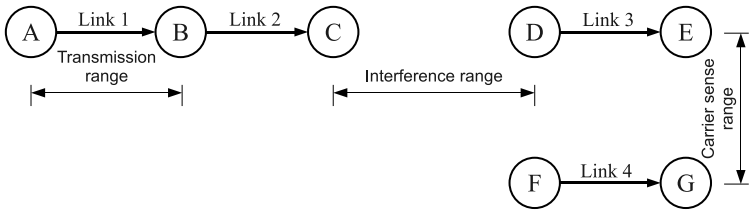


Figure 4.1: A network topology graph.

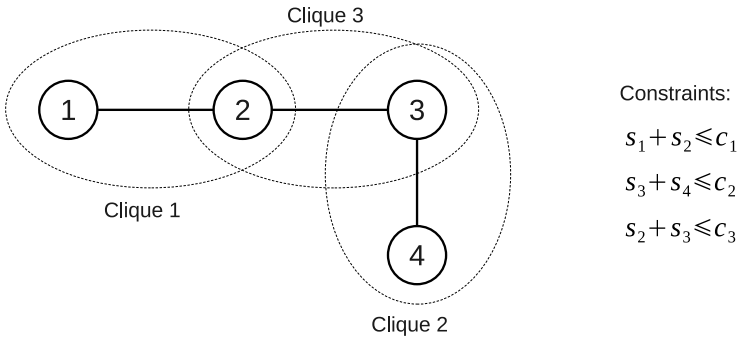


Figure 4.2: A contention graph for the sample network topology (binary interference model)

cliques. Any pair of links in the same maximal clique is prohibited from transmitting concurrently in order to avoid collisions. Thus, for each clique there is a resource constraint, in terms of air time, represented by the clique capacity. Fig. 4.2 shows three maximal cliques for our sample network, with corresponding resource constraints, where s_l is the sending rate of link l, and c_j is the capacity of clique j. Clique capacities and rates are typically normalized to a value between zero and one. Clique capacities are usually between 0.8 and 0.9, depending on the efficiency of the MAC protocol.

In Fig. 4.2, links belong to the same clique either because of contention or interference. Link 1 and link 2 belong to clique 1 because node B cannot send and receive at the same time. Link 2 and link 3 belong to clique 2 because signals from node D may completely corrupt those from node B to node C given the binary interference model. Node D and E are within carrier sense range, and thus link 3 and link 4 belong to a same clique.

The binary interference model is widely critiqued for its over-conservative scheme. Concurrent transmissions that could have been allowed are prohibited in the model, causing the net-

work to be under-utilized. A recent measurement study reveals that interference is typical and *partial* [6]. This means that transmissions from an interfering node may corrupt only a fraction of the packets received at a remote node. As we will demonstrate in following sections, modeling interference as contention results in a misleading optimization problem, where capacity may be wasted and actual receiving rates may be far from fair.

In addition to the over-conservative scheme, the binary interference model requires enumerating maximal cliques in a given graph, which is a well-known NP-hard problem. Particularly, it is extremely difficult to even approximate the maximal cliques in dense graphs. With the binary interference model, links within interference range of each other cannot be active concurrently. As a result, the contention graph might be dense, because there might be a large number of interferers to a remote node. Existing graph-based proposals adopt approximations in deriving distributed rate control algorithms. The work presented in [34] assigns links within two hops to the same clique, however this is a very conservative scheme and the communication overhead and delay in forwarding control information might be significant. Other works use the perceived level of collisions in the neighborhood as a indication to the saturation of clique capacity [40, 28], however they require modifying underlying link layer protocols.

4.2 Objective Functions

The objective function of an optimization problem measures the performance of a network given choices of rate allocations. Typical objective functions seek to maximize the overall utility in the network. The higher the overall utility, the better a rate allocation is. A typical objective function looks like

$$\max f(s) = \sum_{l \in L} w_l U(s_l),$$

where s is the vector of sending rates for links, L is the set of links in the network, and U is the utility function. Wireless links are basic units in this objective function. Alternatively, we can

define an objective function as

$$\max f(s) = \sum_{t \in T} w_t U(s_t),$$

where s is the vector of source rates for flows, and T is the set of flows in the network. The basic units become transport-layer flows in this case. We will further contrast the choices between link-based and flow-based formulations in the following sub-section.

In general, the overall utility is the weighted sum of the individual utility of basic units that participate in the rate allocation in the network. Utility functions are usually defined over the transmission rates allocated to the basic units to quantify their "happiness" given a rate allocation. The weights provide the flexibility of treating units differently according to certain criteria, however many research works simply consider all the basic units equally, and use a global weight of 1. In this dissertation, we set transmission weights for each link equal to the number of flows traversing the link. This approximates a flow-based rate allocation in a link-based problem.

In addition to maximizing the overall utility, an optimization problem should also maintain a sense of fairness between competing basic units. However, there is a tradeoff between utility and fairness. Should the fairness be ignored, the maximal overall utility could be achieved by simply assigning all the air time to those basic units with highest rates, but paying the price of starving those slower units. Some units may have to lower their rates in order to be fair with other competing units. Unfortunately, determining the fairness is a non-trivial job. Interesting questions include: 1) Should we keep the absolute equality between competing units as the topmost criteria, or allow some extent of unfairness as long as its for the good of the overall utility? 2) If we seek absolute fairness, how much is the overall utility sacrificed? 3) Under what scenarios should a unit lower its rate? and 4) How much it should lower the rate if necessary. The answer to the first question is determined by the subjective goal of an optimization problem, but we need systematic methods to seek answers for the rest questions.

An optimization problem uses a specific definition of the utility function to achieve some desired attribute of fairness. The mapping between the alternatives of fairness and their corresponding utility functions is well established in the literature, such as the work presented in [22, 23].

Explicitly, the utility functions are defined as

$$f_\alpha(x) = \begin{cases} logx & \text{if } \alpha = 1 \\ (1-\alpha)^{-1}x^{1-\alpha} & \text{otherwise.} \end{cases} \tag{4.1}$$

We can achieve proportional fairness when $\alpha = 1$, or max-min fairness when $\alpha \to \infty$. By intuition, max-min fairness seeks the absolute fairness between competing basic units. From the above definition we can see that even a minor difference in the allocated rate will lead to enormous changes in the utility when α takes a large value. In contrast to max-min fairness, proportional fairness emphasizes maximizing the overall utility. The $\log(\cdot)$ form of the utility function suggests that increasing the rate of a basic unit from an nearly-starved rate gains more than increasing that from an already high enough rate in terms of the utilities. The "optimal" rate allocation of an optimization problem varies according to the specific target fairness sought.

With the binary interference model, the utility functions are defined over the *sending* rates of the basic units. This is because both the sending and the receiving side of a transmission have to be interference and contention free according to the model, and the rates are the same on the two sides if we also ignore the inherent loss of wireless transmissions. As we have briefly pointed out, the impact of interference is partial in practice, and the receiving rate of a transmission could be significantly lower than its sending rate as a result. With the partial interference model proposed in this dissertation, the utility functions should be based on the *receiving* rates of the basic units.

4.3 Basic Units in Resource Allocation

In an optimization problem, the rate allocation could be conducted either over wireless links or transport-layer flows. A wireless link is a unidirectional sender-receiver pair of nodes in a wireless mesh network. A transport layer flow may cross multiple wireless links from the source to the destination, while a wireless link may have a few traversing flows. Both link-based and flow-based formulations are commonly adopted in the literature. Flow-based formulations correlate more closely to user perceived experience, and are more favorable in most modeling scenarios than

V	The set of vertices (nodes) in the network.
L	The set of wireless links in the network.
s_l	The sending rate of link l.
r_l	The receiving rate of link l.
d_l	The delivery ratio of link l.
A	The vector of interference factors in the network, whereas a_{il} represents the interference factor of link i interfering with link l.
$I(l)$	The set of wireless links that interfere with link l.
$F(l)$	The set of links that link l interferes with.
C	The set of maximal cliques in the contention graph.
$L(j)$	The set of links in maximal clique j.
c_j	The effective capacity of maximal clique j. Typical values are around 0.85, minimizing the chances of transmission collisions.
$C(l)$	The set of maximal cliques that contain link l.
T	The set of transport-layer flows in the network.
$T(l)$	The set of flows traversing link l.
s^t_{src}	The source rate of flow t.
r^t_{end}	The end receiving rate of flow t.
s^t_i	The sending rate of flow t at hop i.
r^t_i	The receiving rate of flow t at hop i.
$h(t)$	The length of flow t in hops.
$k(l,t)$	Hop k of flow t, corresponding with link l.
$U(\cdot)$	The utility function for each link or flow.
w_l	The weight for link l.
w_t	The weight for flow t.

Table 4.1: Notations used in the formulation

link-based formulations. However, little work in the literature discusses their intrinsic differences and scenarios where one is more favorable than another. In this subsection, we use two example formulations to unveil the reasons behind their differences. For ease of our discussion, we first define important notations in Table 4.1.

In our comparison, we consider the problem of maximizing the sum of utilities, $U(\cdot)$, over all the basic units in a wireless mesh network, with constraints typically imposed by link capacities. For simplicity, we use the binary interference model in this section.

The typical form of a link-based optimization problem looks like:

$$\mathbf{A}: \text{ maximize } \sum_{l \in L} w_l U(s_l) \tag{4.2}$$

subject to:

$$s_l \geq 0, \forall l \in L, \tag{4.3}$$

$$\sum_{l \in L(j)} s_l \leq 1, \forall j \in C. \tag{4.4}$$

This problem seeks to maximize the sum of all link utilities in the network, with constraint (4.4) regulating that the sum of the link rates within a clique should not exceed its capacity. For simplicity, we use a capacity of 1. Constraint (4.3) simply says that link rates should be non-negative. Problem **A** assumes that links have infinite backlogs.

The typical form of a flow-based optimization problem looks like:

$$\mathbf{F}: \text{ maximize } \sum_{t \in T} w_t U(s^t_{src}) \tag{4.5}$$

subject to:

$$s^i_l \geq 0, \forall i \in T, \forall l \in L, \tag{4.6}$$

$$\sum_{l \in L(j)} \sum_{i \in T(l)} s^i_l \leq 1, \forall j \in C, \tag{4.7}$$

$$s^t_{i+1} = r^t_i, \forall t \in T, 1 \leq i \leq h(t) - 1. \tag{4.8}$$

Constraint (4.7) requires that the overall flow rates in a clique should not exceed its capacity.

Note that a link's rate is equal to the sum rate of its traversing flows, so we have

$$s_l = \sum_{i \in T(l)} s_l^i \tag{4.9}$$

where s_l^i is flow i's rate at link l, and $T(l)$ is the set of traversing flows at link l. With Eq.(4.9), we can cast constraint (4.7) into the same form as constraint (4.4). This makes sense in that the sum of wireless transmissions in a clique should not exceed its capacity no matter whether the optimization problem is link-based or flow-based.

Problem **F** has more constraints than problem **A**. Problem **F** breaks the rate of a flow t into a set of rates at each hop on the flow's path. Constraint (4.8) is necessary for a feasible rate allocation by requiring that a flow is transmitted at the same rate as it is received at each intermediate nodes on the flow's path.

The set of feasible solutions to the problem **F** is a subset to that of problem **A**. A feasible solution to **A** satisfies constraint (4.4), and thus constraint (4.7), but not necessarily (4.8). On the other hand, a feasible solution to **F** must also be a feasible solution to **A**, because the solution must satisfy constraints (4.7) and (4.8), and thus satisfy (4.4). We can easily construct a rate allocation that satisfies **A** but not **F** by assigning different rates to a flow at separate hops.

The objective function of problem **A** and problem **F** are the same if flows in the network are disjoint. Disjoint flows do not share any common links, and this suggests that an active link only has a single traversing flow. If we assume that all links have same capacities, then maximizing a link's rate is equal to maximizing a traversing flow's rate. The solutions to both problems are also the same.

If flows are joint in the network, problem **A** and **F** have different objective functions. **A** maximizes the overall utility over all links in the network, while **F** maximizes the overall utility over all transport-layer flows in the network. This difference suggests that the two models use different criteria in measuring the performance of a network. As we will demonstrate later, **A** may starve some flows in order to have the network better utilized, the behavior of which may appear

to be unfair from **F**'s perspective.

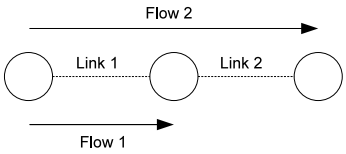
Problem **A** seeks to maximize the overall network utility. Problem **A** only considers link rates, and prevents under-utilized links. As a result, starving some flow in the network won't affect the problem performance as long as the links are occupied by some other flows. This attribute, however, may cause unrealizable rate allocations in practice, or unfairness between flows in the network. In the following discussions, we will use two examples to further demonstrate these two problems with link-based optimization problems.

Problem **F** aims to maximize the sum of flow utility in the network. As compared to problem **A**, **F** cares more about the user-perceived network performance, because flow rates are closely correlated to user's experience. If flows are joint in the network, **F** will increase a flow's rate only when 1) there is extra capacity available on the flow's path so that no other flow's rate needs to be decreased, or 2) the benefit of increasing a flow's rate is more than the loss of decreasing the rate of another joint flow. The utility function takes a $\log(\cdot)$ form for proportional fairness. The rate of change in the utility decreases as the flow rate increases. Decreasing the rate of a fast flow and increasing the rate of a slow flow might increase the overall flow utility in the network. **F** avoids starving flows, because a flow's utility approaches negative infinity as its rate becomes zero.

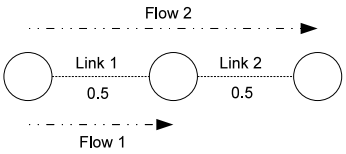
We can also combine problem **A** and **F** by using **A**'s objective function with **F**'s constraints. The optimization problem of this combination also seeks to maximize the overall network utility, but with constraints for realizable flow rates in the problem solution. However, the objective function also ignores flow rates. As a result, some flows may still be starved in the optimal rate allocation of the problem.

We now use two network topologies, illustrated in Fig. 4.3(a) and 4.4(a), as examples to our discussions above. In both topologies, there are two links with two flows traversing them. Along with the network topologies, we also present the problem solutions for the topology when using problem **A**, problem **A**'s objective function with **F**'s constraints, and problem **F** respectively.

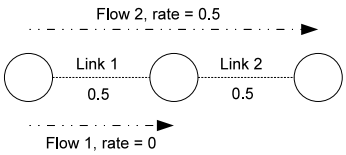
For ease of our discussions, we treat all the basic units equally by assigning their weights,



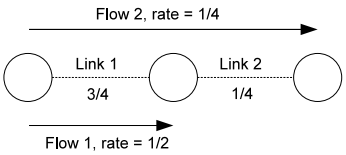
(a) Network topology



(b) The optimal solution for problem **A**.



(c) The optimal solution for problem **A** constrained by realizable flow rates.



(d) The optimal solution for problem **F**.

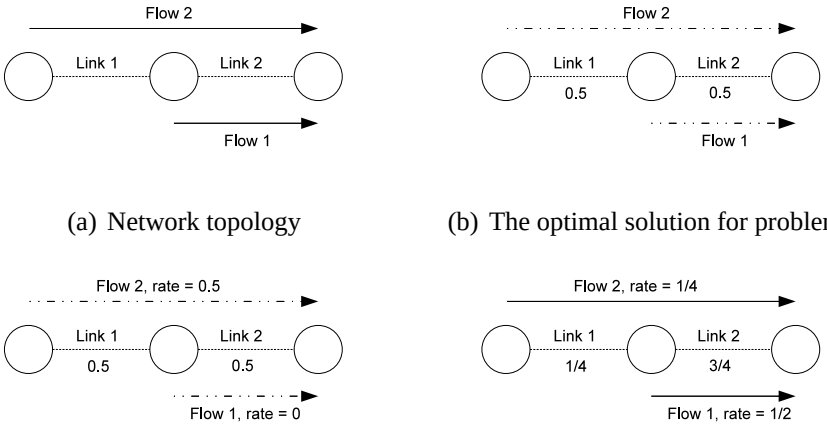
Figure 4.3: Example network topology 1

w_l or w_t, to 1 in problem **A** and **F**. We use proportional fairness, and the utility function takes a $\log(\cdot)$ form. We assume that a wireless link equally splits its allocated bandwidth over its traversing flows. We also assume that all links have an identical capacity of 1, and cliques have a capacity of 1 as well. Link and flow rates are presented in terms of the normalized rate as compared to the link capacity.

For the network presented in Fig. 4.3(a), problem **A** assigns both link 1 and 2 with a rate of 0.5 in the optimal solution. This is because the two links cannot be active concurrently as they share the node in the middle of the topology. With this optimal solution, the network is fully utilized and the two links are treated equally. Note that **A** ignores the flow information, and that's why we use dotted arrows for flows in Fig. 4.3(b) and 4.3(c).

The optimal solution from **A** causes un-used link capacity at link 2 in practice. **A** assumes infinite backlogs for all links, however this is not true in the example network. The optimal solution assigns link 1 with a rate of 0.5, suggesting that both flow 1 and 2 have a rate of 0.25. In practice, the second hop of flow 2 is not able to send faster than it receives, so the actual rate for link 2 is 0.25 as well, even though we use problem **A**.

Fig. 4.3(c) presents the optimal solution when we use the objective function of **A** with



(a) Network topology

(b) The optimal solution for problem **A**.

(c) The optimal solution for problem **A** constrained by realizable flow rates.

(d) The optimal solution for problem **F**.

Figure 4.4: Example network topology 2

constraints of **F**. Both links still have the same rate of 0.5 so that the network is fully utilized, which is the very objective of **A**. With **F**'s constraints, the second hop of flow 2 has to send at the same rate as it receives. As a result, flow 1 is starved, and flow 2 gets all the capacity. If flow 1 is assigned with any fraction of link 1's capacity, link 2 has to be under-utilized because the first hop of flow 2 gets a rate of less than 0.5. Note again, starving a flow won't hurt **A**'s performance as long as the link capacity is filled with some other flows.

As illustrated in Fig. 4.3(d), problem **F** assigns $\frac{1}{2}$ of the link capacity to flow 1, and $\frac{1}{4}$ to flow 2. Given this solution, link 1 gets a rate of $\frac{3}{4}$, and link 2 gets $\frac{1}{4}$. In this network, the two hops of flow 2 and flow 1 have to share the total link capacity of 1. So we have $s_1^2 + s_2^2 + s_1^1 = 1$. The two hops of flow 2 must transmit at the same rate, so $2s_1^2 + s_1^1 = 1$, and thus $s_1^1 = 1 - 2s_1^2$. Note that **F** seeks to maximize the overall flow utility. Given proportional fairness, it's easy to see that **F** looks for the solution that maximizes the product of s_1^1 and s_1^2, which is $s_1^2(1 - 2s_1^2)$. We can obtain the solution by making the derivative of the term equal to 0.

We present the second example in Fig. 4.4(a). Similar to the first example, the optimal solutions with different optimization problems are illustrated in Fig. 4.4(b), 4.4(c), and 4.4(d) respectively.

Problem **A** causes packet drops for flow 2 in practice. The first hop of flow 2 sends at a rate of 0.5 on link 1, however, its second hop can only send at 0.25 because both flows have to share link 2, which is assigned with a rate of 0.5 in the optimal solution. Flow 2's packets will be dropped at the node in the middle.

The optimal solutions presented in Fig. 4.4(c) and 4.4(d) follow the similar idea as discussed in the first example.

In most scenarios, problem **F** generates a more favorable fair rate allocation than **A** does, in terms of both realizable flow rates and fairness. Consider a network similar to Fig. 4.3(b), but with flow 1 spanning only link 1, and two one-hop flows 2 and 3 sharing link 2. **A** will still assign both links with a rate of 0.5 in the optimal solution. However, flow 2 and 3 have to share the same fraction of capacity, 0.5, as flow 1. This allocation is unfair to flow 2 and 3, and the situation could become arbitrarily worse given more flows sharing link 2.

On the other hand, problem **F** is usually limited more by complexities in modeling as compared to **A**. Protocol designers would like to have optimization problems that are convex, so that standard techniques can be applied to derive distributed algorithms [41]. Flow-based problems usually run into more constraints as compared to link-based formulations, and thus could bear complexities that make the problem non-convex. We will show later in this dissertation that flow-based models lead to non-convex problems if partial interference is incorporated. Protocol designers have to use approximations to solve non-convex problems, the solutions of which may not be as straightforward or efficient as they would be in link-based formulations.

Despite the disadvantages, allocating based on links may be preferable in some situations, such as when opportunistic routing causes several paths to be used simultaneously [42]. In opportunistic routing, packets of a flow may take different routes towards the destination as the next-hop receiver is opportunistically determined. It is difficult to construct a flow-based optimization problem when each packet may take a different path.

Chapter 5

Partial Interference Model

We present our partial interference model in this section. The proposed model promotes more accurate optimal rate control designs than the binary interference mode in the sense that it incorporates the impact of partial interference, which is prevalent in wireless mesh networks. In the proposed model, an interfering node may corrupt a fraction of the frames received at a remote node. Partial interference is not represented in the contention graph, but is instead represented in a directional interference map and incorporated as an additional constraint or as part of the objective function. Similar to discussing the binary interference model, we now present the partial interference model from the perspectives of resource constraints, objective functions, and link-based vs. flow-based formulations.

5.1 Resource Constraints

To model partial interference accurately, we separate contention constraints from interference constraints. Contention is represented as an undirected edge between two vertices (links), and interference is modeled as a directional edge from the interfering link to the receiving link that is affected by the interference. The modified contention graph corresponding to Fig. 4.1 is shown in Fig. 5.1. Maximal cliques are then determined as before. Clique constraints, as shown in the figure for Clique 1 and Clique 2, are the same as in the binary interference model, but the constraint between links 2 and 3 is modeled separately.

Interference relationships impose constraints on the receiving rates, as illustrated in the figure, where r_l is the effective receiving rate of link l, s_l is the sending rate of link l, d_l is the

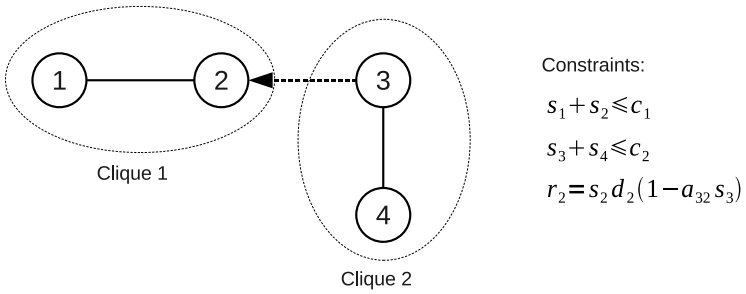


Figure 5.1: A contention graph for the sample network topology (partial interference model)

inherent loss of the link (e.g. due to obstacles or noise), and the term $(1-a_{il}s_i)$ is the loss due to interference from an interfering node i. This constraint is taken from a recent measurement study of wireless mesh networks showing that partial interference can be modeled as a linear function [6]. The interference factor a_{il} represents the degree of partial interference inflicted by the interferer. It is in the range $0 \leq a_{il} \leq 1$, and is unidirectional, meaning that a_{il} may be significantly different from a_{li}. Interfering factors can be experimentally measured between any pair of links in a network by methods suggested in [6, 29], constructing an interference map for the network. A study shows that interfering transmissions are independent of each other, and the joint impact of interferers to a receiving node is merely the product of their isolated impacts [6].

The partial interference model is less conservative than other graph-based approaches because more links are assumed to transmit concurrently. For example, consider links 2 and 3 in Fig. 5.1, and suppose link 3 corrupts 40% of packets received at link 2. If both links transmit at the clique capacity 1, then the sum of their effective receiving rates becomes $1+(1-0.4)=1.6$. In the binary interference model, these links would not be able to transmit at the same time because they would be considered to be in the same clique, resulting in a total effective receiving rate of 1. The partial interference model thus allows for significantly higher utilization of the network.

It is important to recognize that even complete interference cannot properly be modeled as contention. That is, a link i will not become a contender to a remote link l even if the interference factor $a_{il}=1$. Consider again the relationship between links 2 and 3 in Fig. 5.1. Suppose $a_{32}=1$. If interference is modeled as contention, then both links will transmit at a rate of 0.5. However, the

total effective receiving rate will be $r_2 + r_3 = (0.5)(0.5) + 0.5 = 0.75$. With the partial interference model, it is easy to see that link 2 should send at the full rate regardless of link 3's rate. If link 3 continues to send at a rate of 0.5, then the total effective receiving rate will be $(1)(0.5) + 0.5 = 1$. Thus the partial interference model will result in higher utility.

5.2 Objective Functions

When modeling partial interference, it is more accurate to optimize over receiving rates, because the sending and the receiving rates may be significantly different. Based on a recent study [6], we can model the receiving rate of a link by multiplying the individual interference factors:

$$r_l = d_l s_l \prod_{i \in I(l)} (1 - a_{il} s_i). \tag{5.1}$$

For link-based formulations, the objective function becomes

$$f(r) = \sum_{l \in L} w_l U(r_l),$$

where r is the set of effective link receiving rates. For flow-based formulations, the objective function is

$$f(r^t_{end}) = \sum_{t \in T} w_t U(r^t_{end}),$$

where r^t_{end} is the receiving rate at the end of the flow. We will further discuss the calculation of r^t_{end} in a following subsection.

The multiplicative term that arises when modeling the effect of overall interference on a receiver may make the optimization problem non-convex, sacrificing well-established techniques in solving convex problems. We will show in subsequent discussions that a link-based formulation that incorporates partial interference is still a convex problem if proportional fairness is used. We will also demonstrate that flow-based formulations lose their convexity when considering partial interference.

We are thus faced with a tradeoff between convexity and accuracy when modeling partial interference. A flow-based formulation achieves realizable rate allocations, with a sense of fairness that is more closely correlated to user experience in a network, but at a price of losing convexity in the optimization problem. A link-based formulation, on the other hand, can be convex, but the derived rate allocations may not be realizable for a set of flows, and may give users a sense of unfairness, especially when several flows traverse the same link in the network, or when some flows have more hops than others.

5.3 Link-Based Problem Formulation

We consider the problem of finding an optimal rate allocation that maximizes the sum of link utilities in a wireless mesh network, which consists of a set L of stationary links. We use the following assumptions in our formulation.

- Contention between links is binary (either fully contending or not at all) and symmetric. Existing work suggests that the contention between neighboring transmissions is asymmetric and time varying [6]. We focus on exploring the impact of partial interference to optimal rate control in this dissertation, and leave the more accurate modeling of contention in future research work.
- Links have infinite backlog of frames to send. We address the scenarios when links do not have enough frames to send in the implementation of the proposed algorithms.
- The impact of interference from links are independent and linear with respect to the interferer's sending rate, as described in (5.1).

Given a contention graph with maximal cliques C and an interference map A, the optimization problem maximizes the sum of link utilities, which are functions of link *receiving* rates, in a wireless mesh network:

$$\mathbf{P}: \max_{s} f(r) = \sum_{l \in L} w_l U(r_l) \tag{5.2}$$

subject to:

$$s_l \geq 0, \ \forall l \in L, \tag{5.3}$$

$$r_l = d_l s_l \prod_{i \in I(l)} (1 - a_{il} s_i), \ \forall l \in L, \tag{5.4}$$

$$\sum_{l \in L(j)} s_l \leq c_j, \ \forall j \in C. \tag{5.5}$$

We assume the utility function U of a link is continuously differentiable, strictly concave, monotonically increasing, and approaches negative infinity as the argument approaches zero from the right.

Problem **P** is non-convex because of the multiplicative term in constraint (5.4). However, the problem can be reformulated by substituting (5.4) into the objective function. Depending on the utility function, the problem may or may not be convex. If we seek to maximize network utility while maintaining proportional fairness, then we let $U(\cdot) = \ln(\cdot)$, and the problem is convex. The objective function becomes

$$f(s) = \sum_{l \in L} w_l \left(\ln s_l + \ln d_l + \sum_{i \in I(l)} \ln (1 - a_{il} s_i) \right). \tag{5.6}$$

Note that the terms can be reordered and that maximizing (5.6) gives the same optimal rates whether or not the delivery ratios d_l are considered, so that the objective function may be reformulated as

$$f'(s) = \sum_{l \in L} w_l \left(\ln s_l + \sum_{i \in F(l)} \ln (1 - a_{li} s_l) \right). \tag{5.7}$$

This suggests that the explicit values for d_l are irrelevant to the optimal rates of the given optimization problems. Thus, we can reformulate problem **P** as a convex problem $\mathbf{P}'$

$$\mathbf{P}' : \max f'(s) \tag{5.8}$$

subject to:

$$s_l \geq 0, \ \forall l \in L, \tag{5.9}$$

$$\sum_{l \in L(j)} s_l \leq c_j, \ \forall j \in C. \tag{5.10}$$

Each link l is associated with a weight w_l in the formulation. As we have discussed in the previous section, link-based models may generate rate allocations that are not realizable . In chapter 7, we will demonstrate how to use link weights to achieve realizable rate allocations for traversing flows.

5.4 Non-Convexity in the Flow-Based Formulation

The flow-based formulation differs from the link-based formulation in that it maximizes end receiving rates of multi-hop transport-layer flows, where only the source link of each flow has an infinite backlog.

The optimization problem for this formulation is

$$\mathbf{Q} : \max_{s} f(r) = \sum_{t \in T} w_t U(r^t_{end}) \tag{5.11}$$

subject to:

$$s^t_k \geq 0, \ \forall t \in T, \ k = 1, \ldots, h(t), \tag{5.12}$$

$$s^t_k = r^t_{k-1}, \ \forall t \in T, \ k = 2, \ldots, h(t), \tag{5.13}$$

$$\sum_{l \in L(j)} \sum_{t \in T(l)} s^t_{k(t,l)} \leq c_j, \ \forall j \in C, \tag{5.14}$$

where each r^t_k is a function of sending rates, according to (5.4).

Constraint (5.13) makes problem **Q** non-convex. This constraint arises because in this

formulation there is no longer an assumption that all links have an infinite backlog of packets to send, as degradation and interference at an earlier hop causes the next hop to have less data available to send. In order to achieve realizable rate allocations for flows, we require that a flow is transmitted at the same rate as it is received at each hop, otherwise congestion or starvation would occur along its path.

Because this problem is non-convex, a distributed solution may not be as straightforward or efficient. An interesting future research work might be exploring approaches to relax the condition in (5.13), and thus casting the problem in a form similar to to $\mathbf{P}'$.

5.5 Distributed Algorithm

We derive a distributed algorithm to solve problem $\mathbf{P}'$, based on the methods presented in [41]. This problem meets Slater's condition [43], giving us strong duality. We seek to solve the problem in a distributed fashion by finding the solution to the dual using Lagrangian relaxation. The Lagrangian of problem $\mathbf{P}'$ is

$$\begin{aligned} L(s,\lambda) &= f'(s) + \sum_{j\in C} \lambda_j \left(c_j - \sum_{l\in L(j)} s_l \right) \\ &= f'(s) - \sum_{j\in C} \sum_{l\in L(j)} \lambda_j s_l + \sum_{j\in C} c_j \lambda_j \\ &= f'(s) - \sum_{l\in L} s_l \sum_{j\in C(l)} \lambda_j + \sum_{j\in C} c_j \lambda_j \\ &= \sum_{l\in L} g(s_l,\lambda) + \sum_{j\in C} c_j \lambda_j, \end{aligned}$$

where λ_j are Lagrange multipliers for constraints (5.5) and

$$g(s_l,\lambda) = w_l \ln s_l + \sum_{i\in F(l)} w_i \ln(1-a_{li}s_l) - s_l \sum_{j\in C(l)} \lambda_j. \tag{5.15}$$

Note that $g(s_l,\lambda)$ is concave in s_l and approaches $-\infty$ to the left and right, so that for a

given λ there is always a unique maximizer

$$\bar{s_l}(\lambda) = \arg\max_{s_l} g(s_l, \lambda). \tag{5.16}$$

This can easily be found by taking the derivative of g with respect to s_l and setting it equal to zero:

$$\frac{w_l}{\bar{s_l}} - \sum_{i \in F(l)} \frac{w_i a_{li}}{(1 - a_{li}\bar{s_l})} = \sum_{j \in C(l)} \lambda_j. \tag{5.17}$$

We can use efficient algorithms, such as Newton's method, to solve for the optimal rates according to (5.17). Define $h(s_l)$ such that

$$h(s_l) = \frac{w_l}{s_l} - \sum_{i \in F(l)} \frac{w_i a_{li}}{(1 - a_{li}s_l)} - \sum_{j \in C(l)} \lambda_j. \tag{5.18}$$

According to Newton's method, we can approach $\bar{s_l}$ over iterations by

$$s_l(k+1) = s_l(k) - \frac{h(s_l(k))}{h'(s_l(k))}. \tag{5.19}$$

Note that this iterative calculation is conducted within each node, and the CPU overhead is trivial as compared to that incurred from exchanging the control messages over the wireless antenna.

The dual function to problem $\mathbf{P}'$ is given by

$$\begin{aligned} Z(\lambda) &= \max_{s} L(s, \lambda) \\ &= \max_{s} \sum_{l \in L} g(s_l, \lambda) + \sum_{j \in C} c_j \lambda_j \\ &= \sum_{l \in L} \max_{s_l} g(s_l, \lambda) + \sum_{j \in C} c_j \lambda_j \\ &= \sum_{l \in L} g(\bar{s_l}(\lambda), \lambda) + \sum_{j \in C} c_j \lambda_j, \end{aligned}$$

and the dual problem is

$$\mathbf{D} : \min Z(\lambda) \tag{5.20}$$

subject to:

$$\lambda \succeq 0. \tag{5.21}$$

We use the gradient projection method to iteratively obtain the optimal λ for the problem. From Danskin's theorem [44], we know that

$$\begin{aligned} \frac{\partial Z}{\partial \lambda_j} &= \frac{\partial}{\partial \lambda_j}\left[f'(s) + \sum_{i \in C} \lambda_i \left(c_i - \sum_{l \in L(i)} s_l \right) \right]_{s=\bar{s}} \\ &= c_j - \sum_{l \in L(j)} \bar{s_l}. \end{aligned} \tag{5.22}$$

Using a step size γ in the negative direction of the gradient gives the algorithm

$$\lambda_j(k+1) = \max\left(0, \lambda_j(k) - \gamma(c_j - \sum_{l \in L(j)} \bar{s_l}(k))\right), \tag{5.23}$$

where

$$\bar{s_l}(k) = \bar{s_l}(\lambda(k)).$$

The convergence of the algorithm is well established in the literature, even when it is asynchronous [41]. Once λ converges to the optimal solution, λ^*, of the dual problem, the optimal solution, s^*, to the primal problem is given by

$$s^* = \bar{s}(\lambda^*).$$

Chapter 6

Numerical Results

We seek to determine in what situations the partial interference model outperforms binary interference models, and by how much. We use MATLAB to numerically compute solutions to the rate optimization problem for several different wireless networks. We use network topologies that represent basic situations — these can be thought of as building blocks out of which larger topologies can be formed.

We introduce three binary interference models that we compare with the partial interference (PI) model. The interference-as-contention (IC) model replaces any interference mappings with contention, no matter how small the interference factor a. The interference-ignored (II) model simply ignores any interference mappings and models only contention. The adaptive contention (AC) model follows the IC model or the II model, depending on which model has higher performance. Thus the AC model gives binary contention the benefit of the doubt — it ignores interference when this provides good performance and models it as contention otherwise.

6.1 Performance metric

To compare these different models, we define a performance metric that is based on the objective function of the PI model, using receiving rates. We justify this by recognizing that receiving rates are what ultimately matters for users of the network. Data that is sent but is lost due to interference is not considered useful. Thus the comparison should be made between the performance observed with PI-derived receiving rates r^* and the receiving rates r' actually obtained by the other model from its sending rates s', according to the PI constraint on receiving rates.

For ease of interpretation, we consider the ratio R of performances P, that is,

$$R = P(r^*)/P(r'). \tag{6.1}$$

Thus, the comparison will simply read that the PI model outperforms the other model R times.

The PI model uses proportional fairness, so that its objective function[1] is

$$f(r) = \sum_{l \in L} \ln r_l. \tag{6.2}$$

However, scores obtained from $f(r)$ range from $-\infty$ to zero, making it non-intuitive to ascertain how significant a better score might be in comparison to a worse score. We introduce the performance function

$$P(r) = e^{f(r)/|L|}, \tag{6.3}$$

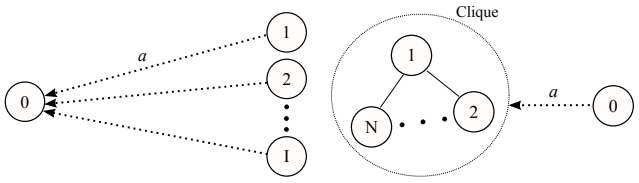
where $|L|$ is the number of links in the network. Note that if $f(r^*) > f(r')$, then clearly $P(r^*) > P(r')$, maintaining the ordering of feasible rate vectors r, based on the objective function scores. Furthermore, note that P turns out to be the geometric mean of receiving rates, which ranges between zero and one, and is normalized with respect to the size of the network. Therefore, we study the ratio R of performances, as denoted in (6.3), between the PI model and other models for various network topologies.

6.2 Results

We consider three generic network topologies and plot R for each topology and for each contention model being compared with the PI model. Each topology is represented in the figures as a combined contention graph and interference map, according to the PI model. Clique capacities in each topology are all $c = 0.85$.

In all cases, the IC model never does as well as the PI model because modeling interference

[1]For simplicity purpose, we ignore link weights w_l in this section.



(a) I links interfering with a single link.

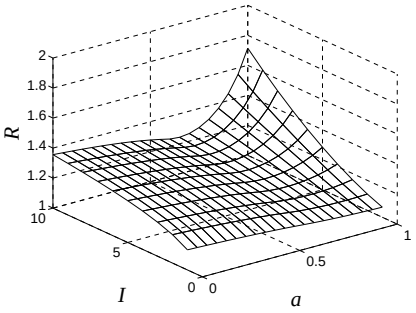
(b) N contenders in a clique with one interferer.

Figure 6.1: Topologies used for numerical results.

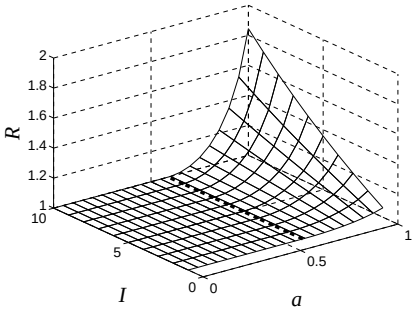
as contention is too conservative. For low values of interference, it is better to let links send at faster rates and suffer some packet loss. At high values of interference, it is better to have the interfered link send at a faster rate than the interferer, to provide better performance and fairness. However, modeling interference as contention is often better than ignoring it when interference is high. Thus in most cases, the combined AC model follows the II model for low values of interference and follows the IC model for high values of interference.

Fig. 6.1(a) shows the first topology, where I links interfere with a single link with a common interference factor a, but do not interfere with each other. Fig. 6.2(a), 6.2(b), and 6.2(c) plot R for this topology for the PI model against the IC, II, and AC models, respectively. The dotted curves show where R begins to be greater than one. Interestingly, the PI model and the II model perform exactly the same for values of a below 0.59. This is because, for low values of a, the cost of interference is offset by the gain of the interferer sending at full capacity. Thus, both the PI model and the II model calculate sending rates at full capacity for each link. For larger values of a and I, the PI model outperforms the binary interference models more than 1.5 times.

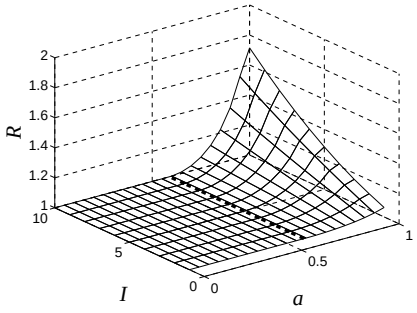
Fig. 6.1(b) shows the second topology, where a single link has interference factor a on N links that contend in a single clique. Fig. 6.3(a), 6.3(b), and 6.3(c) plot R for this topology for the PI model against the IC, II, and AC models, respectively. The dotted curves show where R begins to be greater than one. The PI model starts performing better than the II model at much lower values of a when N is large. This is due to the fact that the contending links already have small rates as a consequence of sharing the medium. Utilities are lowered much more by interference when sending rates are small. Thus, even for low values of a, the PI model does not calculate



(a) Ratio R of performance between the PI model and the IC model.

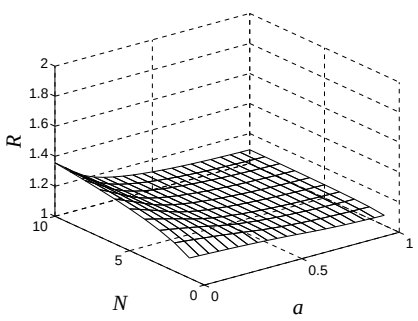


(b) Ratio R of performance between the PI model and the II model. The dotted curve marks where R begins to be greater than one.

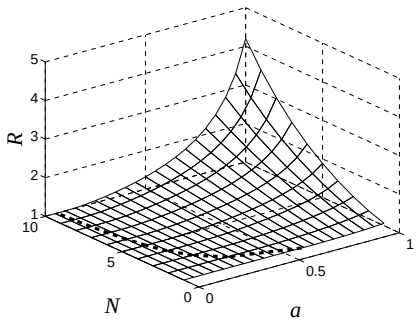


(c) Ratio R of performance between the PI model and the AC model. The dotted curve marks where R begins to be greater than one.

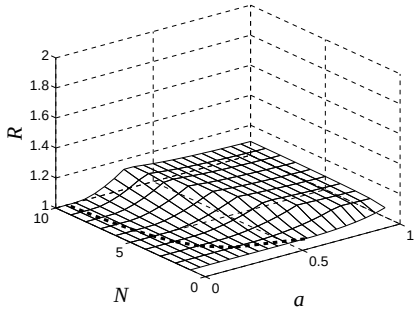
Figure 6.2: Numerical results for I interferers on one link.



(a) Ratio R of performance between the PI model and the IC model.



(b) Ratio R of performance between the PI model and the II model. The dotted curve marks where R begins to be greater than one.

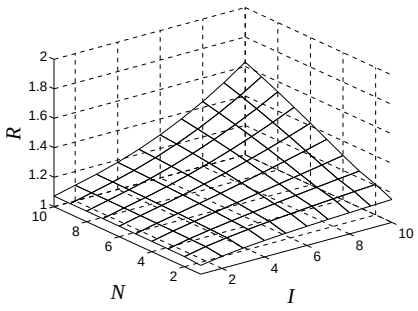


(c) Ratio R of performance between the PI model and the AC model. The dotted curve marks where R begins to be greater than one.

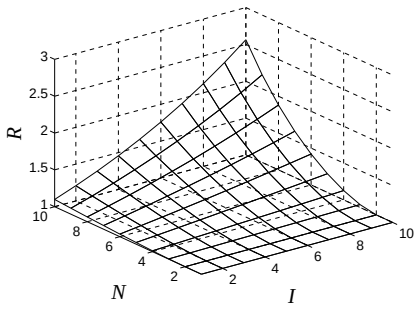
Figure 6.3: Numerical results for N contenders with one interferer.

sending rates at full capacity. However, for higher values of a and N, the PI model outperforms the IC model only about 1.1 times.

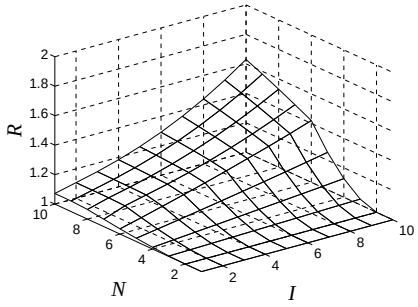
To demonstrate the worth of the PI model, we consider a topology combining features of the first two, where I links have a fixed interference factor $a = 0.4$ on N links that contend in a single clique. Fig. 6.4(a), 6.4(b), and 6.4(c) plot R for this topology for the PI model against the IC, II, and AC models, respectively. Experimental results show that it is typical for interference factors to range anywhere between zero and one in a real network, with usually at least one interferer on a link having a factor of at least $a = 0.8$, so choosing $a = 0.4$ in this topology is a somewhat conservative comparison [6]. The combined effect of several interferers and several contenders causes the PI model to perform significantly better than the binary interference models.



(a) Ratio R of performance between the PI model and the IC model.



(b) Ratio R of performance between the PI model and the II model.



(c) Ratio R of performance between the PI model and the AC model.

Figure 6.4: Numerical results for I interferers and N contenders, with $a = 0.4$.

Chapter 7

Protocol Implementation

In addition to the numerical results, we implement the rate control algorithms in an experimental wireless mesh network and examine the protocol performance. As discussed in Section 2, the practical performance of a distributed algorithm might be significantly worse than its expected performance in the theoretic modeling. In this section, we discuss implementation details and practical concerns when implementing the rate control algorithms.

7.1 Implementation Goals

Various engineering approaches are available to implement the fair rate control algorithms. We use the following criteria when making implementation decisions:

- Ease of development. The protocol implementation should allow the ability to rapidly prototype, deploy, and evaluate the fair rate control protocols. As compared to a kernel-space implementation, a user-space design is preferable because it is easier for developers to write and manage code with less rigid programming requirements and with a wider range of development tools at their disposal.

- High performance. Interference and contention interactions between wireless transmissions is significantly affected by transmission power and bit rates, the proposed fair rate control should be able achieve typical high speed bit rates supported by the IEEE 802.11 standard family so that the experimental results collected in the testbed settings are consistent with real world deployment. In our experimental evaluation, the fair rate control algorithms should work when nodes communicate at a bit rate of 54 MBit/sec.

- Flexibility in supported network protocols. The fair rate control algorithms should work with a wide range of transport layer protocols, such as TCP variants or UDP, and the IEEE 802.11 link layer standards to facilitate exploration of various options in solving the performance problems for wireless mesh networks.

We examined toolkits in the literature that could be potentially adopted to implement the rate control algorithms. The Click modular router [45] serves the most relevant purpose, but the methods used in the toolkit do not meet our implementation criteria in terms of ease of development and high performance. Click is able to achieve high performance by using its kernel module, however this module polls devices and device drivers must be modified to support polling. Its user-space module, on the other hand, uses a packet capture library [46] that requires setting the wireless card in the promiscuous listening mode, which is known to suffer from low data rates and be susceptible to packet drops. Given these considerations, we do not select Click as the development toolkit for this dissertation.

In this dissertation, the rate control algorithms are implemented as components of a user-space toolkit called WiFu that is being developed at BYU to support experimental wireless transport protocols. In the following sections, we first briefly introduce the experimental testbed setup, and then discuss implementation details for the network interference and contention map measurement and for the fair rate control algorithms.

7.2 Wireless Mesh Testbed

We build an experimental wireless mesh testbed in the computer science department building of BYU. The testbed consists of 28 nodes that are placed at the first and second floors of the building. Fig. 7.1 illustrates nodes on the first floor, and those on the second floor are placed to provide a similar network topology and wireless coverage.

Each node is a desktop computer with a 2.4 GHz Pentium processor and 767 MByte of memory, and runs on Linux operating system. Nodes communicate with each other through two network interfaces: a 100 MBit/sec Ethernet card and an IEEE 802.11 a/b/g wireless card. The

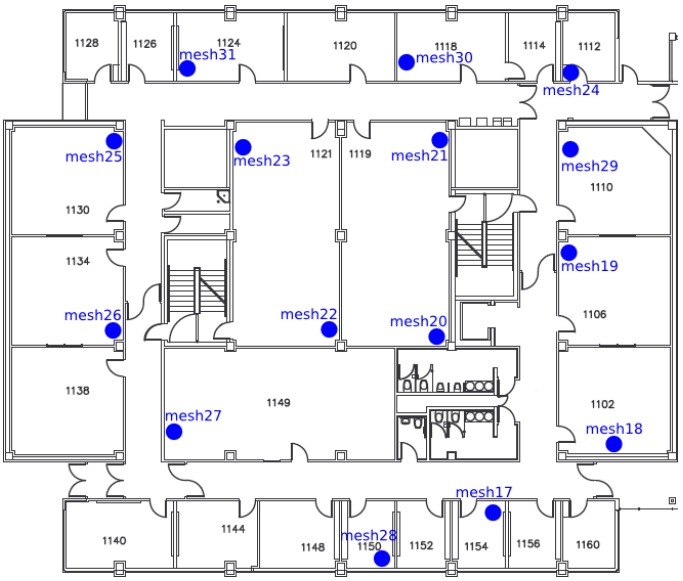


Figure 7.1: Wireless mesh testbed first floor nodes

Ethernet interface is typically used to exchange scheduling messages between a separate server that manages the experiments and the participating mesh nodes.

7.3 Network Interference and Contention Map Measurement

Our network interference and contention map measurement is inspired by [6], however the measurement steps used in the work are not applicable to our implementation. The contention map measurement relies on frequent communications with the device driver. During the process of exploring a user-space approach, we also find that the broadcast-based method used in the work may generate inaccurate results because of the significantly different behaviors between broadcast and unicast transmissions.

The contention map in [6] is measured using the Click modular router [45]. In particular, the transmission feedback feature in Click provides a transmission report for each packet, such as the transmission was ACK-ed successfully, retried to the maximum and dropped, etc. This feature allows each sending node to directly measure the transmission rate using broadcast, and

thus measure its access to the wireless medium without the need of other receivers. If a pair of neighboring nodes are within carrier sense range of each other, their transmission rates should vary from broadcasting alone to concurrently.

Our investigation also reveals that measuring the interference factor using broadcast may generate inaccurate results. Wireless transmissions via broadcast may experience significantly different performance from unicast in throughput and packet loss rate, and thus the measured interference factor might fail to accurately reflect the impact of partial interference in experiments. This observation may also invalidate the contention map measurement in [6] if the exact value of contention matters in a fair rate control algorithm, such as the one proposed in [47], because the contention map is measured using broadcast but nodes typically communicate through unicast in the experiments. More details about our investigations on broadcast vs. unicast will be discussed in the chapter of experimental results.

We use unicast transmissions to measure contention between a pair of neighboring nodes, say node A and B. The measurement is performed in four steps:

1. Sending nodes select receivers. Each sending node selects a receiver to form two separate unicast links, say A→C and B→D. The receivers should be chosen in a way that transmissions from one sending node do not interfere with the packet reception of another link. This is to separate the impact of interference from contention.

2. Sending nodes take turns to transmit data to their respective receivers using unicast at full link capacity. Receivers calculate the transmission rate. In the example, C calculates R_a and D calculates R_b.

3. Sending nodes concurrently transmit data to their respective receivers using unicast at full link capacity. Receivers calculate the transmission rate. In the example, C calculates R_a^b and D calculates R_b^a. The node in the superscript is the potential contending node.

4. Receivers calculate the contention ratio. For node C, the contention ratio is defined as $\frac{R_a^b}{R_a}$, and D calculates the ratio as $\frac{R_b^a}{R_b}$. Note, the two contention ratios could be different for asymmetric

contion. Node B contends with node A if $\frac{R_a^b}{R_a} < 1$, and similarly node A contends with B if $\frac{R_b^a}{R_b} < 1$.

The above steps are repeated for each permutation of nodes of interest for contention measurement.

In this dissertation, a pair of nodes is considered to be contending with each other if either one of the contention ratios is less than 1, because the partial interference model assumes symmetric contention between a pair of neighboring nodes. For the same reason, the exact contention ratio value is also ignored in the fair rate control algorithm derivation. The algorithm proposed in [47] incorporates asymmetric contention and requires the exact contention ratio information.

The interference map is measured using a similar unicast approach. However, the interfering node should be within the interference range of the receiver and outside of the carrier sense range of the sender of the interferee link. For example, given a interferee link A→C, a interfering node B should be within the interference range of C and beyond the carrier sense range of A to separate the impact of contention from interference. To measure interference of node B to link A→C, node A first transmits to C at link capacity while B remains silent, and C calculates the receiving rate from A, R_a. In the next step, both A→C and B→D transmit at link capacity concurrently, and C calculates the receiving rate R_a^b when transmissions from B are present. Node B interferes with link A→C if $\frac{R_a^b}{R_a} < 1$, and the interference factor when B transmits at full link capacity is defined as $\left(1 - \frac{R_a^b}{R_a}\right)$.

Both the contention and interference measurement methods are considered as user-space approximations to the their real values. Because the contention measurement is examined on the receiver's side, it is difficult to precisely distinguish the impact of contention from interference. We have yet to find a more accurate way of measuring contention so that we can determine the exact cause of a change in the contention ratio or interference factor in the measurement. This work is left as future research. In our measurement, the receiver is selected as the node that is most distant from the contending node, and the interferer is selected as the node that is farthest from the sending node of the interferee link.

Our measurement script consists of a server component that schedules the transmissions

and collect results, and a client component that runs on each participating node to receive commands from the server and send back measurement results. The clocks of the server and the mesh nodes are synchronized using the NTP protocol. To minimize possible communication delays, the server side script creates separate threads for each mesh node, and adds enough cushion before the scheduled concurrent transmissions. In order to make the measurement accurate, irrelevant mesh nodes in the experimental testbed are set to non-overlapping frequencies to avoid interference. Any process that may generate wireless traffic, such as the routing process that periodically broadcasts for route updates, are terminated on the participating nodes to ensure only the desired measurement transmissions are sent out in the testbed. Measurement traffic is generated using UDP flows and packets are sent as quickly as possible. Participating nodes operate at the same bit rate and power level as they do in the experiments. The unicast packet size is set to be 1500 bytes, the same as used in the experiments.

Similar to the approach in [6], this method requires *offline* measurements. The measurement results are saved in a text file that is loaded by nodes when the fair rate controller is launched. Recent work proposes an *online* estimation of interfering factors in a wireless LAN [48]. We focus on examining whether the proposed partial interference model delivers the expected accuracy and benefits in practice given a pre-generated interference map, and the offline measurement suffices for this purpose. Note that our proposed protocol can easily adopt the online estimation method given any future necessity.

7.4 Maximal Clique Enumeration

To calculate fair rates, links need to construct their local contention graphs and find the maximal cliques using the network interference and contention map. Unfortunately, enumerating maximal cliques in an arbitrary graph is a well-known NP-hard problem, and the problem is extremely difficult in dense graphs. With the binary interference model, links within interference range of each other cannot be active concurrently. As a result, the contention graph is likely to be dense, because there are likely a large number of interferers to a remote node. Existing graph-based

proposals adopt approximations by either over-conservatively assigning links within 2 hops apart to the same maximal clique [34], or requiring major modifications to underlying link layer protocols [40, 28].

With the partial interference model, it is viable to design efficient and accurate protocols that enumerate maximal the cliques in a contention graph. The work presented in [49] suggests that efficient algorithms exist for enumerating maximal cliques in graphs that are 1) sparse and 2) closed under the operation of taking subgraphs. The maximal cliques in the partial interference model only consist of links with senders within the carrier sense range of each other. In a typical wireless mesh network, the number of immediate contenders is quite limited, and we can assume the contention graph satisfies the above two assumptions. Note that choosing the best clique enumeration algorithm and exploring efficient alternatives are beyond the focus of this dissertation. The specific algorithm adopted is decoupled from our rate control protocol. We simply choose from existing well-established algorithms that suffice our investigation.

For the purpose of this dissertation, we use the Bron-Kerbosch algorithm [50] to calculate all the maximal cliques that a link belongs to. This algorithm takes a link adjacency matrix as input, and uses a recursive method, which is exponential in terms of the computational complexity. However, it is efficient enough for the proposed protocol, because the algorithm only needs to work on a limited number of contenders. In subsequent discussions, we demonstrate a design that further decreases the size of the input to the algorithm. Furthermore, the algorithm is executed within each node, and the CPU overhead is trivial as compared to the wireless communication overhead.

The fair rate control algorithms use a distributed protocol, and links populate and maintain their local contention graphs and maximal cliques. A new sender collects local contention information and populates the maximal cliques in the contention graph in three steps: 1) declares its intention to join the network and requests a list of neighbors from existing neighbors; 2) informs its neighbors of its own list of neighbors; and 3) decomposes maximal cliques in its neighborhood using the information collected from step 1). The second step is necessary so that existing senders are able to update their maximal cliques given the presence of the new sender.

In the first step, the new sender broadcasts to declare its intention to join the network. The message is broadcasted a few times to make sure every neighboring node successfully receives the message, because hidden terminals or other sources of interference may corrupt the message. Upon receiving the broadcast, existing senders respond with their list of contending neighbors. At this stage, existing nodes are not able to determine which maximal cliques the new sender should belong to. They have to wait until receiving the list of neighbors from the broadcasting node. In the second step, the broadcasting node creates its own list of contending neighbors using the responses from the existing nodes, and sends the list back to the responders. Both new and existing nodes use the same steps as described below to update their maximal cliques.

The new sender puts together the contention information and populates maximal cliques using Bron-Kerbosch algorithm. To decrease the input size, we use nodes to represent links in the clique decomposition. Although the maximal cliques should be decomposed over contending links, we use the sending nodes of those links and their contention relationship as input to the algorithm. Once the algorithm decomposes all the maximal cliques, we recover the cliques-of-links from the cliques-of-nodes by replacing each sending node with the set of outgoing links from that node. This delegation is based on the observation that the contention between any pair of links is an effect of their sending nodes. Modeling the contention between links is equivalent to modeling that of their sending nodes. A node is likely to have at least several outgoing links in active networks, so using nodes in the algorithm can decrease the input size to the algorithm.

The proposed clique enumeration process is able to work in real time as compared to previous work in the literature that artificially assigns any nodes within two hops to a same maximal clique but still suffers from excessive fair rate convergence time [34]. Our clique enumeration code is implemented as a function in the fair rate control protocol. With the partial interference model, a node only initiates a new enumeration process when it receives notification of a contender joining or leaving the network. Given the benefits of communicating to a less number of contenders rather than much more potential interferers, our protocol implementation is able to afford frequent message exchanges among contending nodes at a level of 100 milliseconds, and is able to collect

necessary information for clique enumeration within a few hundred milliseconds. Experimental results show that nodes are able to converge to the optimal fair rates within one and half seconds, which makes our fair rate control protocol practical to use in real networks.

7.5 User-space Development Toolkit

The rate control protocols are implemented in WiFu, a software toolkit that is being developed by the Internet research lab of BYU. The toolkit provides the capability of user-space development of rate control protocols for wireless mesh networks, promoting easy and rapid development. WiFu is also able to support high throughput data transmissions so that the experimental results achieved through the toolkit remain close to the performance in real deployment. In a performance evaluation, WiFu is capable of saturating a 54 MBbit/sec wireless link. The PI, IC, and II modeling altervatives are implemented as components in WiFu.

Wifu allows a developer to intercept packets at every node on the path of a flow in wireless mesh networks. Packet interception is achieved by using the Linux **iptables** software with the **netfilter** interface [51]. **netfilter** is a Linux kernel extension that allows applications to intercept packets according to a chain of **iptables** rules that are set by the user and specify actions to handle them. **iptables** provides a feature that stores the intercepted packet to a queue. WiFu can register a handler for packets stored in the queue and read packets from the queue to apply rate control or reliability actions, which typically include modifying the header of the packet, transmitting the packet at the desired time, or discarding the packet. The chain specification used in our experiments take following forms:

```
INPUT -i wlan0 -p udp –dport 5000:5100 -j NFQUEUE –queue-num 0

OUTPUT -o wlan0 -p udp –dport 5000:5100 -j NFQUEUE –queue-num 1

FORWARD -i wlan0 -p udp –dport 5000:5100 -j NFQUEUE –queue-num 2
```

The first rule is used to intercept UDP packets that arrive at the node as their destination via the wlan0 interface, which is the wireless card interface in our experiment setup. The dport

option specifies that only packets of flows with the port number falls in the provided range will be captured. This allows WiFu to only employ rate control on the desired set of flows. Captured packets are added to **iptables** queue 0, and WiFu can register an INPUT handler to retrieve packets stored in this queue. The second rule is used to intercept packets that leave the node as their source, and the captured packets are added to queue 1. The third rule is for packets that traverse the node as an intermediate hop on their flow path.

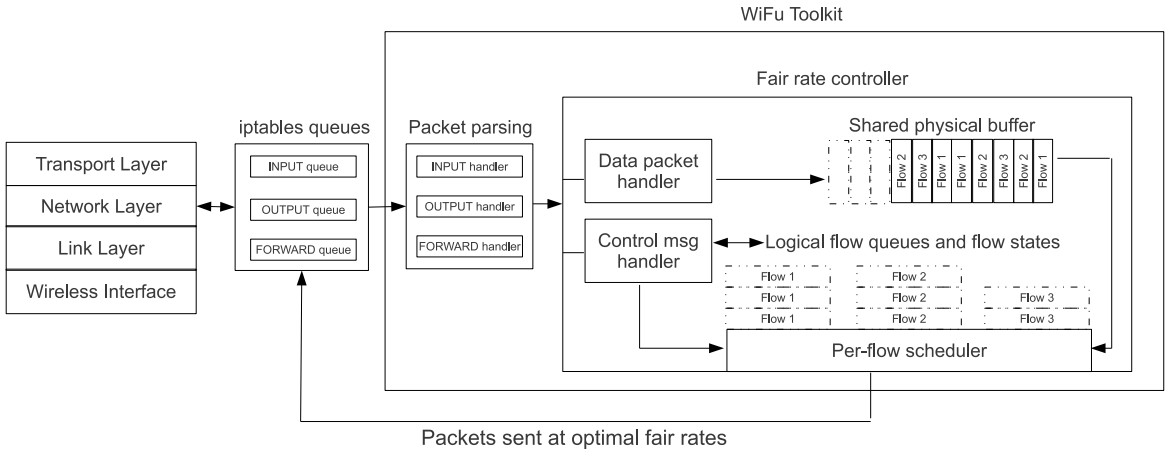


Figure 7.2: Fair rate control implementation architecture

Fig. 7.2 presents an architectural view of the WiFu toolkit and the fair rate control algorithm implemented as its component. WiFu registers a handler for each **iptables** queue. The handler is responsible for parsing the captured packets to retrieve the piggybacked fair rate control message if any and recover the original IP packet. Both the data packet and control message are then passed to the fair rate controller. The rate controller stores the packets in a physical buffer shared by all the traversing flows. Each flow is associated with a state structure, containing statistical information of the flow, such as the logical flow queue length, the instantaneous incoming and outgoing rate of flow packets, the calculated fair rate for the flow, etc. The sum of each flow queue length should be less than the overall physical buffer space. The control message handler is the "brain" of the rate controller. It manages the control message exchange sequence with other nodes, and

calculates flow fair rates according to the rate control algorithms, such as PI, IC, or II model in this dissertation, by using the information from other nodes and local flow states. The per-flow scheduler manages packet transmissions according to the fair rate of the flow.

7.6 Fair Rate Control

In order to implement practical rate control, we need to consider: 1) what should be the weight of link l, w_l, in Eq. (5.17); 2) what to do should the infinite backlog assumption become invalid in practice; and 3) how nodes exchange control messages in an timely and efficient manner.

To overcome the limitations in link-based models, as discussed in Section 4.3, we use the number of traversing flows as the link weight. Links with more traversing flows will have heavier weights in the optimal rate control, and thus be assigned more bandwidth. To prevent unfairness between flows sharing the same link, each link equally divides the assigned bandwidth to its traversing flows. Nodes maintain soft flow-state for each traversing flow. A flow state is dynamically created by a node upon receiving the first packet of the flow, and purged after a period of time without any new packets of the flow. A node also needs to inform its neighbors should a link weight change so that the neighbors adjust the weight parameters in Eq. (5.19).

The assumption of an infinite backlog in link-based models may not hold in practice, particularly for multi-hop flows. Even though we use link-based modeling to achieve the desirable convexity in the optimization problem, the protocol implementation implicitly bears the constraint on realizable flow rates, i.e. Eq. (5.13). In the optimization problem, links should transmit at their derived rates to calculate clique prices according to Eq. (5.23). However this may not be achievable in practice. A multi-hop flow goes through multiple nodes, and an upstream node may experience heavier contention than a downstream one, which might not have enough flow packets to send in this scenario. Using the calculated fair rate in this case, senders may mistakenly think that they are converging to the optimal rate and the link capacity is well utilized. In fact, they may send at much lower rates, and the wireless channel only appears to be well utilized, leaving idle channel capacity.

To deal with this problem, our protocol implementation provides the option to use either the actual transmission rates, or the calculated fair rates when calculating clique prices according to Eq. (5.23). For multi-hop flows, the first option is suggested. If a sending node transmits at a lower rate than expected due to insufficient packets, the sum rate of the clique becomes lower than the optimal target, and the clique price becomes lower, suggesting all links in the clique to increase their rates. In this way, flows with sufficient packets are able to utilize the idle channel. However, the rate allocation is unable to converge to the optimal target, because there is always a gap between the actual and the optimal transmission rate because of the insufficient packets.

To minimize bandwidth consumption, nodes have the option to opportunistically piggyback control messages on whichever packets that are sent to the desired node. A significant fraction of the control messages can be piggybacked because a multi-hop flow usually has packets sent to the upstream and downstream nodes, which are also contenders to the current node. When piggybacked on a regular IP packet, the control message is inserted as a shim structure between the IP header and the transport protocol header. For neighbors with disjoint flows, control messages are exchanged via explicit messages.

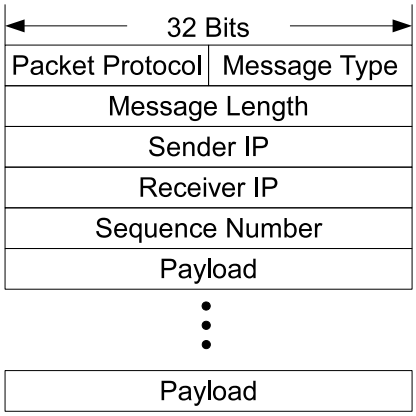


Figure 7.3: Fair rate control message format

Fig. 7.3 illustrates the format of the fair rate control messages. The packet protocol field

saves the original content in the IP header **protocol** field, which defines the data portion of the IP packet, such as TCP or UDP. The field in the IP header has to be changed to indicate the fair rate control message should it be inserted. Upon the IP packet is retrieved, the WiFu handler recovers the **protocol** field of the IP packet and removes the inserted fair rate control message according to its message length field, which indicates the total length of the control message. The message type specifies the type of the control message. The <sender IP, receiver IP> pair uniquely identifies the link for which the control message serves. The sequence number is used for periodically exchanged control messages, such as the node transmission rate updates, so that neighboring nodes update their rates synchronously. The definition of the payload field is subject to the message type.

Control Message Name	Message Usage
MSG_CONTENDER_JOIN	Declare contender join event
MSG_CONTENDER_LEAVE	Declare contender leave event
MSG_CONTENDER_RATE	Declare current node transmission rate
MSG_CONTENDER_NEIGHBORS	Declare current contender list
MSG_CONTENDER_FLOWS	Declare number of traversing flows
MSG_INTERFEREE_INFO	Declare interferee join event
MSG_INTERFEREE_LEAVE	Declare interferee leave event

Figure 7.4: Fair rate control messages

Fig. 7.4 summaries the name of the rate control messages and their usages. The control messages are divided in two categories: the contention related messages and the interference related messages. The contention related messages are exchanged among contenders. For the PI and II model, the "contending nodes" include the set of nodes that are within carrier sense range of each other. For the IC model, it includes nodes that are either within carrier sense or interference range of each other. The MSG_INTERFEREE set of messages are sent from an interferee link to

the interferer node.

MSG_CONTENDER_JOIN is the first message that a node sends out once its very first flow starts. The node declares its intention to join the contention neighborhood by sending this message, and then waits for a period of time, 200 milliseconds in our experiments, for responses from the existing contending nodes. When the waiting time expires, the node starts enumerating the maximal cliques by using the steps described in section 7.4.

MSG_CONTENDER_LEAVE is the last message that a node sends out after all of its flows terminate. Upon receiving this message, active contenders remove the sending node from the list of neighbors, and initiate the contention graph update process, which consists the same steps as the maximal enumeration sequence.

MSG_CONTENDER_NEIGHBORS is used in two scenarios. Upon receiving the join message, existing nodes respond with the MSG_CONTENDER_NEIGHBORS message to inform the new node of their current list of neighbors. The new node puts together all the neighboring information from existing nodes and enumerates maximal cliques in the neighborhood, and then sends out a MSG_CONTENDER_NEIGHBORS message to those responding nodes indicating the list of neighbors of the new node. As the last step of the join event, existing nodes update their maximal cliques by using the list of neighbors from the new node. The message's payload field starts with the number of neighbors, 32-bit, which is followed by the IP address of each neighbors.

MSG_CONTENDER_RATE is periodically exchanged between contending nodes after the join process is finished. The payload field contains the transmission rate of the node. Note, this value is the sum rate of links that originate from this node. This is fine because the sum of the node transmission rates is equivalent to the sum of the link rates in the clique. The node transmission rate is constantly measured in WiFu by using a sliding window. A node sends a MSG_CONTENDER_RATE message every 100 milliseconds. Upon receiving the message, a node updates its clique price according to Eq. 5.23 and then link fair rates according to Eq. 5.19.

A node sends the MSG_CONTENDER_FLOWS message when the number of traversing flows change. Because links are weighted by their number of traversing flows in the optimization

problem. The payload field contains the number of traversing flows of the link.

After a node finishes the join process, it sends a MSG_INTERFEREE_INFO message to its interferers if the PI model is used, so that the interferer starts adjusting the fair rate according to Eq. 5.17. The list of interferers is obtained from the interference map file. The payload field of the message contains the weight of the new link and the interference factor of the interferer to the interferee link. For the similar purpose, a leaving node also sends a MSG_INTERFEREE_LEAVE message to its interferers.

Chapter 8

Experimental Results

We use our implementation to test the performance of the rate control algorithms in our experimental wireless mesh network. This section discusses results and observations from our experiments. We first measure the network interference and contention map using the approach proposed in [6]. We then use our fair rate control framework to calculate link rates using the PI, IC, or II algorithms discussed 6, and control link transmissions according to the calculated fair rates.

The performance of the fair rate control algorithms are evaluated in a partial-interference and then in a contention-only topology, both of which are building blocks for more sophisticated scenarios. Any complex topology in practice boils down to the combination of these two basic scenarios. For the purpose of this dissertation, we only conduct experiments in these two toplogies as they reveal a great deal of insights into how the rate control algorithms perform in practice. Engineering challenges are exposed during our experiments, and are yet to be further investigated before moving forward to explore more complex toplogies. Details of these challenges will be discussed in following subsections.

Overall, our experimental results show that partial interference is prevalent in wireless mesh networks, and modeling interference as contention leads to over-conservative resource allocations. We also observe that measuring the network interference map using broadcast may lead to inaccurate results. A unicast approach is preferrable in terms of measurement accuracy. We also find that the interferee link may not be able obtain higher throughput when the interferer link lowers its transmission rate, due to a non-linear relationship between the two links.

8.1 Experiment Configurations

All the nodes in the mesh network are equipped with a single radio, and operate on the IEEE 802.11 a/b/g standards [52]. This is a general enough network configuration as it represents the most prevailing situation in practical mesh networks today. Although using multiple radios at each node has the potential to achieve higher throughput, there are practical difficulties that arise when attempting to assign non-interfering wireless channels to these radios. Previous work has observed interference between closely-located radios that operate on orthogonal frequencies [53, 54], and the degree of such interference varies over devices of different manufactures. We only consider IEEE 802.11 a/b/g standards, because they are the dominant wireless standards used in practical mesh networks.

In order to make experimental results comparable and repeatable, we configure all nodes in the wireless mesh network with identical settings before running each set of experiments. Of particular interest to this study, we set the following parameters:

- Wireless channel number. This parameter determines the frequency at which wireless signal is to be transmitted. Our research team collaborates on concurrent but unrelated experiments so that they operate at orthogonal frequencies in the network to avoid undesirable interference from each other. Experiments reported in this dissertation run on IEEE 802.11a channel 149.

- RTS/CTS exchange sequence. RTS/CTS is originally designed to solve the hidden-terminal problem, but is known to only partially solve the problem [35] and may degrade overall throughput. As the exchange sequence may alter the interactions between a pair of interferer and interferee links, we set this option to be off.

- Link layer maximum retransmissions. The IEEE 802.11 link layer standard provides the option to retry failed transmissions at the link layer. Enabling this option may overwhelm the impact of partial interference, which is the major focus of this research. We set the maximum retry number to 0.

- Channel bit-rate. The IEEE 802.11 standard family supports multi-rate transmissions. A transmitting node can be configured to adjust its channel bit-rate according to the varying quality of wireless channel. In our optimization modeling, the transmission rate is normalized against link capacity, but altering the channel bit-rate may significantly change the link capacity. We set the channel bit-rate to be fixed at 12 MBits/sec to keep the link capacity constant in the fair rate calculation. Note, our experiments do not directly use the channel bit-rate in actual fair rate calculation, because it includes the overhead of network protocols. Rather the peak throughput of a UDP flow over the link is used.

- Node transmission power. The higher the transmission power, typically the further wireless signals can propagate. There is a tradeoff, however, in choosing the appropriate transmission power for a network. With higher transmission power, more nodes contend with each other. In the extreme case, all nodes in the network contend with each other. On the other side, lower transmission power may generate weak signals, which usually causes unreliable link quality and irreproducible experimental results. We precede our experiments with simple tests to verify that the interference and contention relationships are as planned between nodes of interest. The transmission power for the interferer link is set to 10, which generates significant enough interference to the interferee link, which transmits at power level 9.

Experimental results may still vary significantly over iterations even when running under identical network settings. This is because interference from external and uncontrollable sources is prevalent in the network. To offset those interfering factors, each scenario is repeated for 30 iterations. As results tend to vary in some of the experiments, we use boxplots to report distributions of results in our disscussions. The median is drawn as a short red line in the box, and the upper and lower boundaries are the upper and lower quartile respectively. Outliners, if any, are marked as plus signs. A result is considered to be an outliner if it exceeds 1.5 times of the Inter-Quartile Range (IQR) of results from all the 30 iterations.

We use single-hop UDP flows to generate traffic load in the experiments. For the purpose of this dissertation, UDP is preferred over TCP, because the flow control and congestion control

mechanism of TCP may affect the behavior of link rate controls. Our experiments focus on studying the fundamentals of the rate control algorithms in practice, and the simple and constant rate of UDP flows are more favorable. In our experiments, the interferer and interferee link respectively transmit a large file with the same size of 25 MBytes. The long transmission time makes it easier for us to study how the fair rate converges over updates.

For ease of our discussions, this section presents experimental results that are collected from two topologies, as shown in Fig. 8.1 and 8.14. The first topology is a partial interference scenario, with node 25 causing interference to node 27. The second topology is a contention scenario, with node 25 and 27 contending. Our experimental observations reveal the general interactions in these two basic scenarios, and apply to any topologies that bear similar relationship between nodes in the mesh network. For the partial interference scenario, any topology in which the interferer link has an interference factor of greater than 0.5 achieves similar results to those presented in Section 8.3. The PI model converges to the same link fair rate as the II model does when the interference factor is less than 0.5. For the pure contention scenario, any topology in which senders contend with each other leads to results similar to those discussed in Section 8.4. We have collected results from different sets of nodes in the mesh network to verify the generality of the results discussed in this section.

8.2 Network Interference and Contention Map Measurement

We first discuss experimental results in the network interference and contention map measurement. The proposed partial interference model relies on the interference factor and contention relationship between links in deriving the optimal resource allocation. We show that the broadcast approach used in [6] is inaccrate as compared to the unicast measurements.

During the course of our exploration to find a good user-space measurement, we experiment with a broadcast approach that approximates the steps used in [6]. In this approximation, nodes take turns to broadcast for a pre-specified number of packets, and the interference factor is calculated at the receiving node by comparing the number of packets received with and without

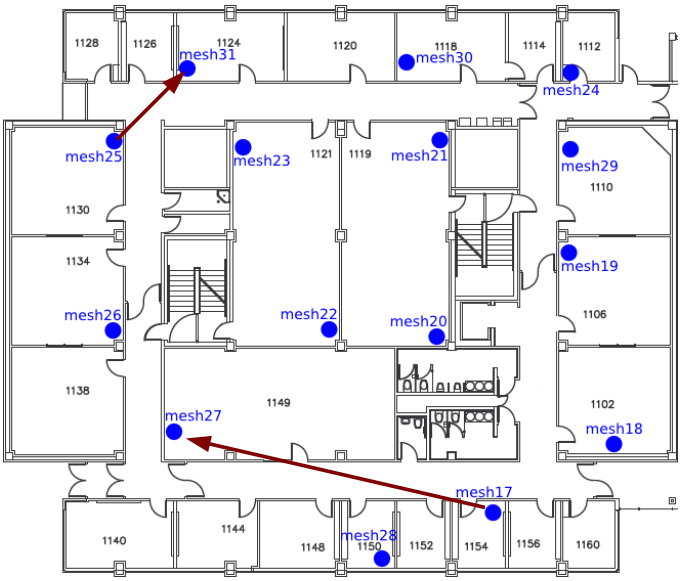


Figure 8.1: Partial interference topology

concurrent transmissions from the interfering node. For example, we want to measurement the interference factor from the interferer node 25 to link 17→27 as illustrated in Fig. 8.1. First, node 17 broadcasts 5000 packets, and node 27 counts the number of packets received, say 5000 packets as an example. Node 17 and 25 then broadcast concurrently, and node 27 counts the number of packets received from 17, say 2200 packets. The interference factor of node 25 to link 17→27 is calculated as $1 - 2200/5000 = 0.56$. Note, node 27 also counts the number of packets received from 25 when receiving the concurrent transmissions, and the number can be used to calculate the interference factor of node 17 to any links that are incoming to node 25. This saves the number of transmissions required to measure all the interference factors of interest.

To measure contention, a node is considered to contend with another if any of its packet is received by the other node. This approach may omit contenders that are beyond transmission range but within carrier sense range of each other.

The measurement results show that interference factors mesured using the broadcast approrach may differ significantly from the actual level of interference experienced by the interferee

link. Fig. 8.2 illustrates the measured interference factor of node 25 to link 17→27 with varying packet sizes. The right most bar in the graph presents the actual level of interference observed in unicast experiments. The long box suggests that the interference from node 25 is strong but also varying over time.

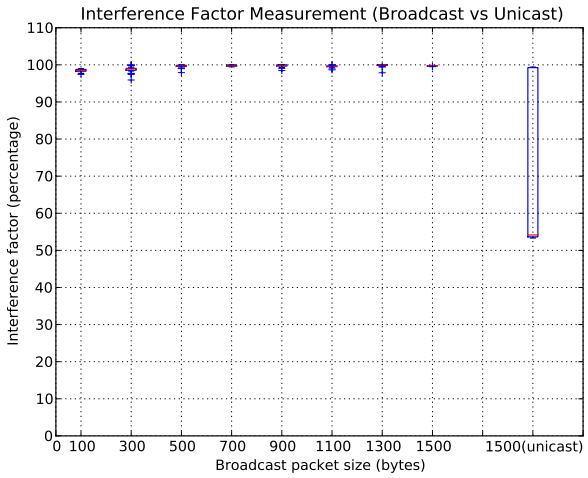


Figure 8.2: Interference factor measurement

The interference factor is also significantly affected by the transmission power of the links. Fig. 8.3 shows measurement results with the transmission power of node 25 lowered to 9 from 10. As compared to the results in Fig. 8.2, the broadcast measurements tend to be more scattered, suggesting the approach is sensitive to packet size when the interference is not as strong.

To further understand how broadcast and unicast differ from each other in network measurement, we compare their performance when transmitting 25 MBytes of data over a one-hop UDP flow. As results plotted in Fig. 8.4 suggest, the broadcast flow suffers from a significantly lower throughput than the unicast flow. The packet loss rate for broadcast transmissions is significantly higher than those of unicast, which explains the low throughput of broadcast flows.

The contention map measurement also generates inaccurate results because of the prevalence of contenders that are beyond transmission range but within carrier sense range of each other.

To avoid the inaccuracy in the broadcast measurement, we use the unicast approach as described in section 7.3 for our experiments. Each possible permutation of participating nodes is

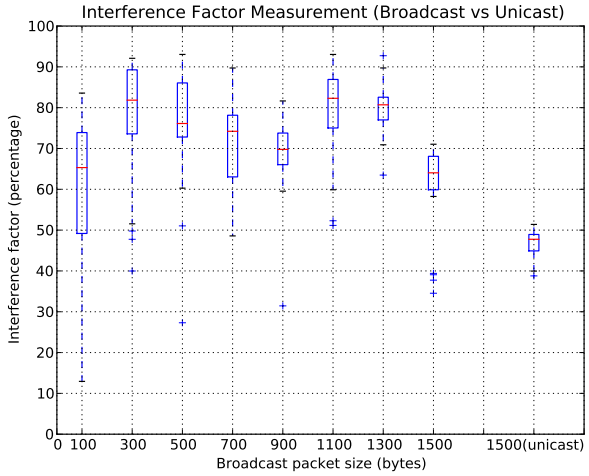


Figure 8.3: Interference factor measurement (transmssion power 9)

explicitly measured over unicast flows. As the rate control experiments are also conducted with unicast flows, this approach achieves more accurate results.

8.3 Partial Interference Scenario

We use a hidden-terminal topology to evaluate the performance of the rate control protocols. As shown in Fig. 8.1, node 25 is the hidden terminal to node 17 when it transmits to node 27. Link 25→31 transmits at power level 10, and link 17→27 transmits at power level 9. Our unicast interference factor measurement suggests that node 25 corrupts approximately 54.6% (median of 30 iterations of measurements) of packets that were sent to node 27 from 17. As illustrated in Fig. 8.2, the interference factor of node 25 to link 17→27 varies significantly over measurements. However, using the median value suffices for our discussions in this section. Because the interferee flow is unable to achieve higher throughput as the interferer flow lowers its transmission rate due to the non-linear interference relationship that is presented shortly.

In following subsections, we discuss the performance of PI, IC, and II models in the hidden-terminal experiments.

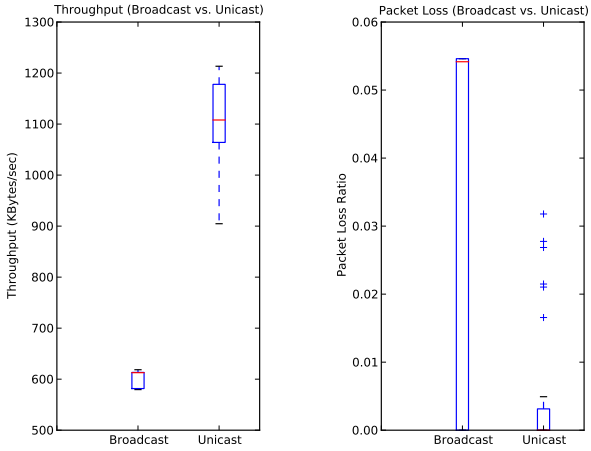


Figure 8.4: Broadcast vs. unicast

8.3.1 Parital Interference Model

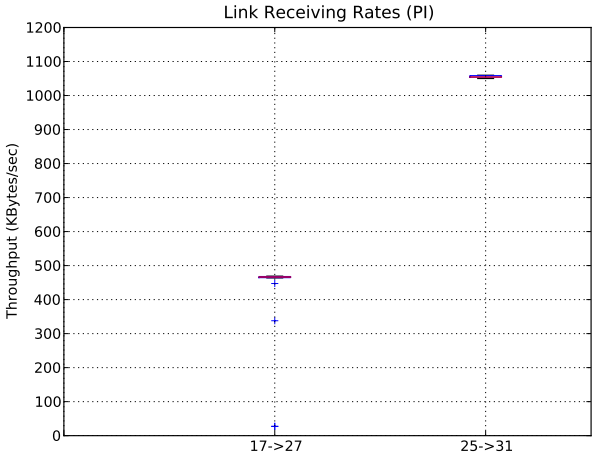


Figure 8.5: Flow throughput (PI)

According to the PI model, node 17 should send at the normalized rate of 1, because it does not contend nor interfere with other nodes in the experiment. Node 25 should transmit at the normalized rate of 0.915, which can be easily derived according to Eq. (5.17) by using $w_l = 1$, $a_{li} = 0.546$, and $\sum_{j \in C(l)} \lambda_j = 0$. Given the link capacity of 1150 KBytes/sec, node 25 should send at about 1052 KBytes/sec, and node 17 should send at 1150 KBytes/sec. In our experiments, the

fair rate controller suggest the correct fair rate to both of the sending nodes.

Fig. 8.5 represents the actual flow throughput achieved in the experiments. The throughput of flow 25→31 matches the optimal fair rate of node 25. The throughput of flow 17→27, however, is significantly lower than the fair rate suggested at node 17. In the worst case, the link suffers badly and receives near-zero throughput.

To determine the reason for this deviation from the theoretical model, we trace events within our WiFu implementation of a rate controller. Our investigation identifies extensive delays between the arrival of outgoing packets at the WiFu framework of node 17. The WiFu framework intercepts the outgoing packets, and passes them to the wireless card driver for actual transmission at the calculated fair rate. Our code generates tracing information, such as time stamp and packet header fields, at the very last and the first statement of the WiFu function that receives the outgoing packets from the system. By calculating the amount of time elapsed between the two time stamps, we obtain the arrival intervals, which characterize the amount of the elapsed time external to the WiFu framework. Possible factors that may cause the delays include system scheduling delays, wireless card driver transmission delays caused by contention or interference, etc.

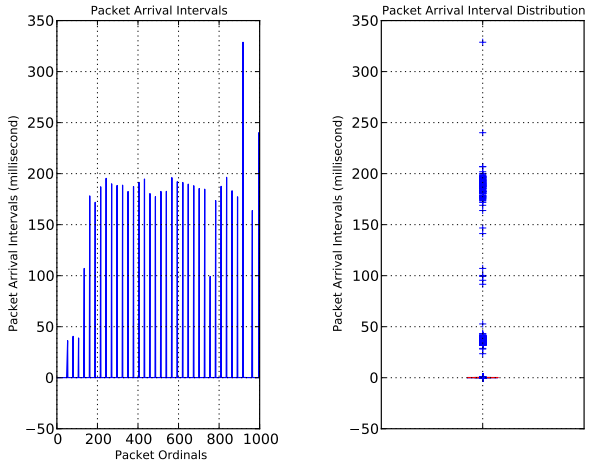


Figure 8.6: Outgoing packet arrival intervals at node 17

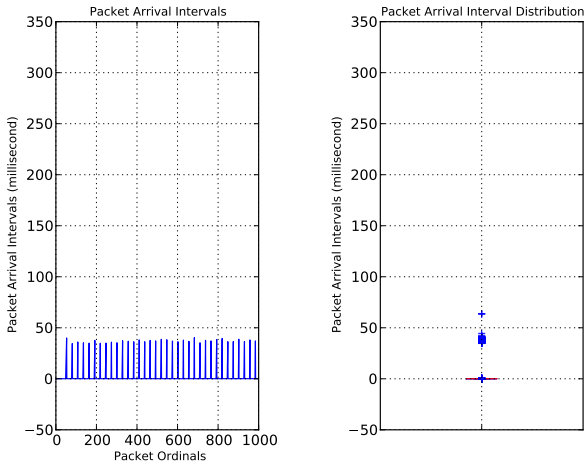


Figure 8.7: Outgoing packet arrival intervals at node 25

Fig. 8.6 and Fig. 8.7 plot the arrival intervals for the outgoing packets of node 17 and 25 respectively. The left half of the figures illustrate the arrival intervals for the first 1000 outgoing packets to reveal more details. The right half of the figures provide the overall distribution of all the arrival intervals. The high spikes in Fig. 8.6 suggest extensively long arrival intervals between outgoing packets at node 17. In contrast, the spikes at node 25 appear to be shorter and more consistent. Note, the majority of the arrival intervals are at the level of 100 nano seconds, and are presented as a line on the x axis in both figures. The fact that spikes occur in both figures suggest that the outgoing packets arrive in bursts. Traces on the sequence number of the UDP packets suggest no packets are dropped before they get into WiFu. The reason that causes the spikes is still unclear to us, and will be one of an important topics of our future research.

Because the interferee is getting less throughput than it should, we test whether a different rate should be used for the interferer. In this investigation, we keep flow 17→27 transmitting at the full link capacity, but gradually increase the transmission rate of flow 25→31 from 0 to 1 at the step of 0.1. As illustrated in Fig. 8.8, the relationship is non-linear, which contradicts the results from [6]. The two lines connect the median values of 30 experiment iterations. We observe similar non-linear pattern between other pair of links in the network.

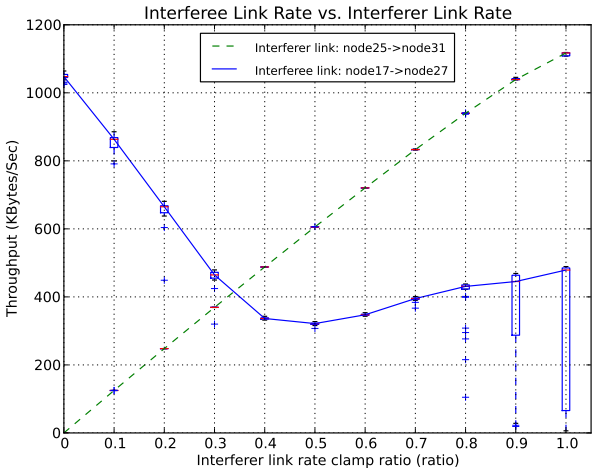


Figure 8.8: Non-linear relationship between interferer and interferee links

Note, the observed non-linear pattern does not necessarily disprove the assumption of linear

impact of the partial interference model. There could be many reasons in practice. One of a possible reasons could be node 25 actually being within the carrier sense range of node 17, even though they are beyond transmission range of each other. To investigate this problem, we need to customize the Clear Channel Assessment threshold in the wireless card driver. This threshold determines the carrier sense range of a node. The customization provides us the ability to vary the carrier sense range of a node, and thus further investigate if node 25 contends with node 17 in this scenario. This piece of work, however, is non-trivial, and belongs to the scope of future work.

8.3.2 Interference as Contention Model

With this model, the interference from node 25 is considered as contention to transmissions from node 17 to 27. As a result, node 17 and 25 should equally share the clique capacity at a fair rate of 0.45, with 0.9 as the effective clique capacity. As shown in Fig. 8.9, both nodes converge to the expected fair rates in about 15 iterations. The two nodes exchange fair rate information every 100 ms. The step size γ is set to 0.5 in updating the clique price λ according to Eq.5.23. The right half of the figure provides more details on the rate convergence during the first 20 updates. Note that node 17 is able to use the full link capacity after node 25 finishes transmission.

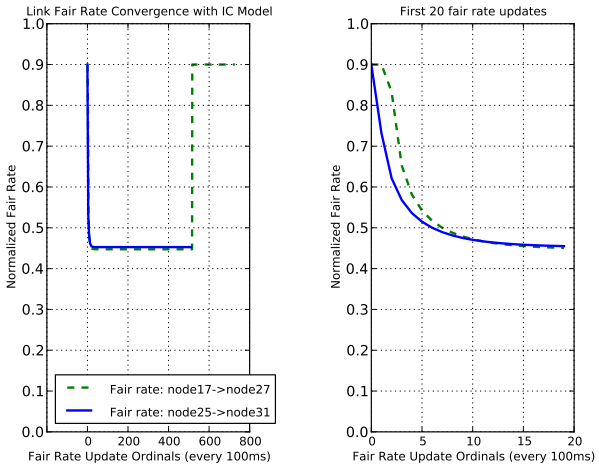


Figure 8.9: Fair rate convergence with IC model

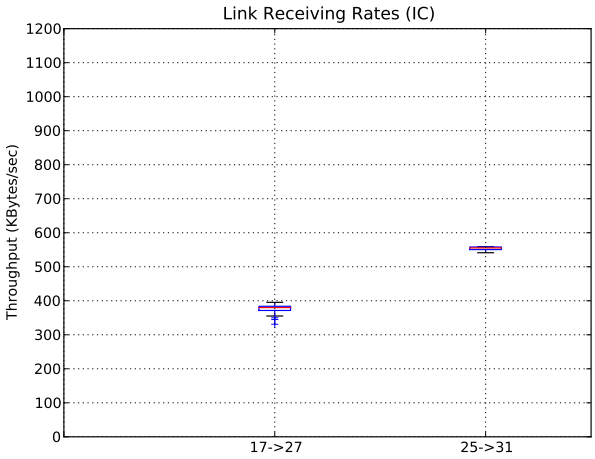


Figure 8.10: Flow throughput (IC)

Fig. 8.10 illustrates the measured throughput for flow 17→27 and 25→31 respectively. Similar to the throughput observed in previous scenario, node 25 is able to transmit at the calculated fair rate, but link 17→27 obtains significantly lower throughput, which is also caused by the extensively high values in outgoing packet arrival intervals.

8.3.3 Interference Ignored Model

In the II model, the interference from node 25 to flow 17→27 is ignored. As a result, both node 17 and 25 transmit at full link capacity. Fig. 8.11 illustrates the measured throughput for flow 17→27 and 25→31 respectively. Consistent with our previous observations, the throughput of flow 17→27 tends to be more scattered and suffers from low throughput. The same extensive high values in outgoing packets manifests at node 17 as well.

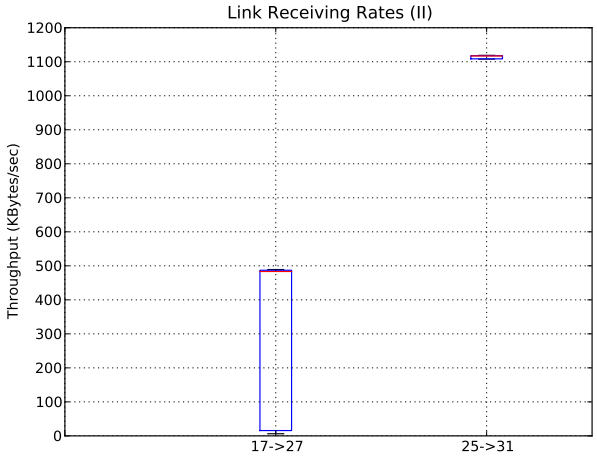


Figure 8.11: Flow throughput (II)

8.3.4 Sum Utility Comparison

We compare the overall performance of the three models in our experiments in this subsection. Recall that the purpose of the fair rate optimization is to maximize the sum utility of all the participating links. In section 6, the performance between different models is compared using the

performance function, which is defined in Eq.(6.3). Fig. 8.12 plots the utility performance values of the three models.

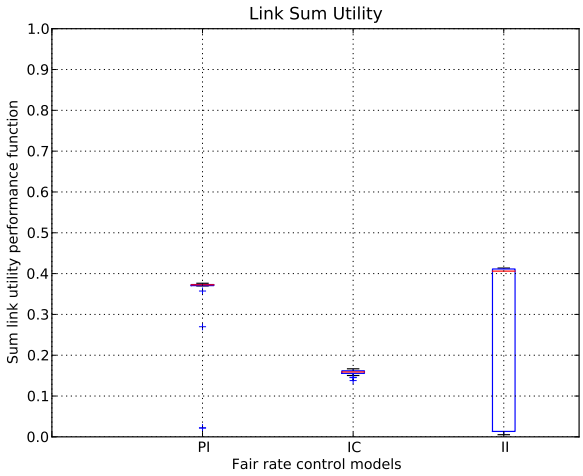


Figure 8.12: Sum utility

The PI model performs better than the IC model in this topology, because IC model is over conservative by treating the interference from node 25 as contention. This confirms our hypothesis that partial interference should be treated differently than contention in the fair resource modeling. As partial interference is prevalent in a wireless mesh network, treating it as contention usually generates over-conservative results.

The II model outperforms the PI model in some experiments, because the interferee link does not benefit from the lowered interferer link rate. On the down side, the II model suffers from more scattered and low throughput results as observed in our experiments, because of the interference from node 25.

Fig. 8.13 reveals the best sum utility that can be achieved in the partial interference scenario. The best sum utility occurs when both 25→31 and 17→27 send at the full link capacity. This is consistent with our previous observations that lowering transmission rate of node 25 does not help 17→27 in most cases. The high IQR when node 25 transmits at full rate is caused by the scattered receiving rate of flow 17→27.

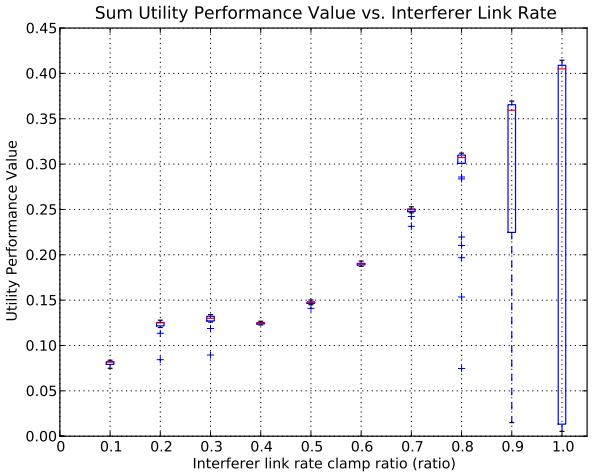


Figure 8.13: Sum utility performance value as a function of interferer rate

8.4 Pure Contention Scenario

We also evaluate the performance of the proposed fair rate control protocol in a scenario with only contention and no interference. As shown in Fig. 8.14, link 27→17 and 25→31 respectively transmit the same amount of data over single-hop UDP flows concurrently. Node 25 and 27 contend with each other.

The PI, IC, and II model behave identically without the interference factor in place. As a result, both node 25 and 27 should send at a fair rate of 0.45, with effective clique capacity of 0.9. Experimental results match the calculated link fair rate as shown in Fig. 8.15 for the PI model. The results for the IC and II model are indifferent from the PI model.

The utility performance of the three models are also indifferent from each other, as shown in Fig. 8.16.

Experimental results from the pure contention topology confirms that the PI model performs no worse than the other two even in a pure contention scenario.

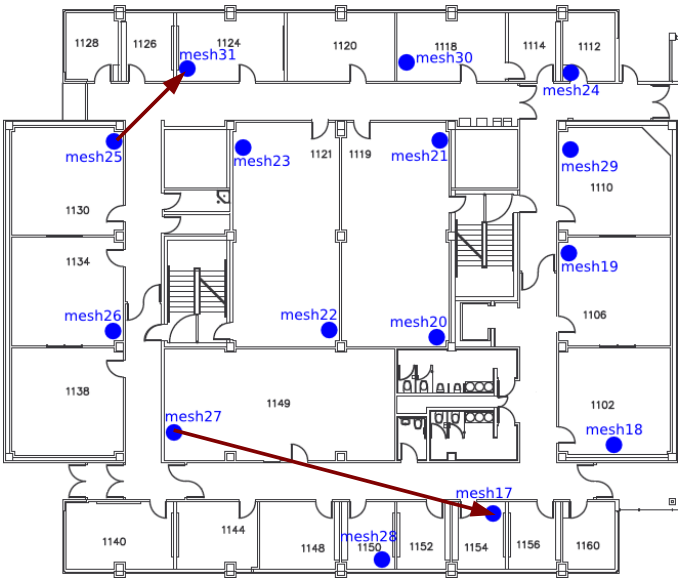


Figure 8.14: Hidden terminal topology

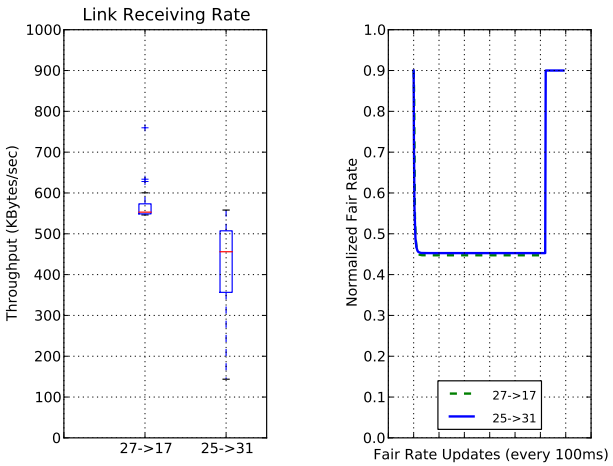


Figure 8.15: Flow throughput and fair rate convergence (PI)

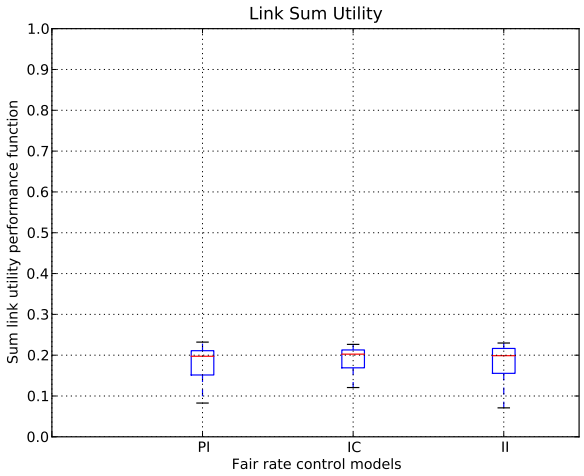


Figure 8.16: Sum utility

Chapter 9

Conclusions

This dissertation contributes to the area of modeling and designing fair rate control for wireless mesh networks in three major perspectives. First, we synthesize models used for optimizing fair rate control for wireless mesh networks, and discuss tradeoffs between different models. Second, we develop a partial interference model [55] that improves the over-conservative binary interference model in the literature. Numerical results show that the PI model outperforms the IC model in all the scenarios, and suggest that partial interference should be modeled separately from contention. Third, we implement the fair rate control algorithm on a mesh test bed. We find that measuring network interference map using broadcast may lead to inaccurate results, and unicast is more preferable for accuracy. Experimental results verify the prevalence of partial interference in a mesh testbed, and show that the partial interference model results in significantly improved performance in sum link utility in a typical interference topology.

Despite the better performance of the partial interference model, we observe a significant deviation between the theoretical performance of the algorithm and the measured performance. This demonstrates a non-trivial gap between theory and practice. In particular, the assumption of a linear relationship between interfering links breaks in our experiments. Lowering the transmission rate of the interferer link does not increase the flow throughput of the interferee link.

This dissertation lays a promising foundation for future research in the area. In particular, the partial interference model can incorporate the asymmetric and time-varying nature of contention, such as the first-principles model proposed in [47]. Further investigation is also needed to find the reason behind the deviation between theory and practice observed in this dissertation. If

the assumption of the linear relationship is disproved, a new model incorporating the non-linear relationship should be proposed. This raises a series of open questions such as whether the new model remains mathematically tractable; whether a distributed algorithm can be derived and implemented, or if not, whether an approximation can be used to achieve close enough results in practice. Additional work is needed to find a balance between complex models and efficient implementations.

References

[1] "One laptop per child project website." [Online]. Available: http://laptop.media.mit.edu

[2] "Technology for all project at Rice University." [Online]. Available: http://tfa.rice.edu

[3] "Bay area wireless user group web site." [Online]. Available: http://www.bawug.org

[4] J. Shi, O. Gurewitz, V. Mancuso, J. Camp, and E. Knightly, "Measurement and modeling of the origins of starvation in congestion controlled mesh networks," in *The 27th Conference on Computer Communications*, 2008, pp. 1633–1641.

[5] R. Govindan, A. Jindal, K.-Y. Jang, S. Rangwala, and K. Psounis, "Understanding congestion control in multi-hop wireless mesh networks," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2008, pp. 291–302.

[6] D. Niculescu, "Interference map for 802.11 networks," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*. New York, NY, USA: ACM, 2007, pp. 339–350.

[7] D. D. Clark, D. P. Reed, and J. H. Saltzer, "End-to-end arguments in system design," *ACM Transaction Computer Systems*, vol. 2, no. 4, pp. 277–288, 1984.

[8] S. M. ElRakabawy, A. Klemm, and C. Lindemann, "TCP with adaptive pacing for multihop wireless networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. New York, NY, USA: ACM, 2005, pp. 288–299.

[9] T. Braun and R. de Oliveira, "A dynamic adaptive acknowledgment strategy for TCP over multihop wireless networks," in *24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 2005, pp. 1863–1874.

[10] A. Helmy, C.-C. J. Kuo, and K. Nahm, "TCP over multihop 802.11 networks: issues and performance enhancement," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. New York, NY, USA: ACM, 2005, pp. 277–287.

[11] F. Wang and Y. Zhang, "Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing*. New York, NY, USA: ACM, 2002, pp. 217–225.

[12] O. Akan and I. Akyildiz, "ATL: an adaptive transport layer suite for next-generation wireless Internet," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 5, pp. 802–817, 2004.

[13] H. Balakrishnan, B. Hull, and K. Jamieson, "Mitigating congestion in wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2004, pp. 134–147.

[14] A. T. Campbell, S. B. Eisenman, and C.-Y. Wan, "CODA: congestion detection and avoidance in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2003, pp. 266–279.

[15] D. E. Culler and A. Woo, "A transmission control scheme for media access in sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2001, pp. 221–235.

[16] R. Bajcsy and C. T. Ee, "Congestion control and fairness for many-to-one routing in sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM, 2004, pp. 148–161.

[17] D. Scofield, "Hop-by-hop transport control for multi-hop wireless networks," Master's thesis, Brigham Young University, 2007.

[18] D. Scofield, L. Wang, and D. Zappala, "A hop-by-hop transport protocol for multi-hop wireless networks," in *The Fourth International Wireless Internet Conference*. HI, USA: IEEE, 2008.

[19] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," *Wireless Networks*, vol. 8, no. 2/3, pp. 275–288, 2002.

[20] X. Yu, "Improving TCP performance over mobile ad hoc networks by exploiting cross-layer information awareness," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2004, pp. 231–244.

[21] V. Anantharaman, H.-Y. Hsieh, K. Sundaresan, and R. Sivakumar, "ATP: a reliable transport protocol for ad-hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*. New York, NY, USA: ACM, 2003, pp. 64–75.

[22] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, 2000.

[23] M. Chiang, D. Kao, T. Lan, and A. Sabharwal, "An axiomatic theory of fairness in network resource allocation," in *29th Annual IEEE International Conference on Computer Communications*, San Diego, CA, USA, 2010, pp. 1–9.

[24] M. Chiang, A. R. Calderbank, J. C. Doyle, and S. H. Low, "Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255–312, 2007.

[25] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *The Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.

[26] P. R. Kumar and V. Raghunathan, "A counterexample in congestion control of wireless networks," in *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. New York, NY, USA: ACM, 2005, pp. 290–297.

[27] S. Shakkottai and Y. Yi, "Hop-by-hop congestion control over a wireless multi-hop network," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 133–144, 2007.

[28] L. Chen, J. Doyle, and S. Low, "Joint congestion control and media access control design for ad hoc wireless networks," in *24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 2005, pp. 2212–2222.

[29] M. K. Han, R. Mahajan, L. Qiu, F. Wang, and Y. Zhang, "A general model of wireless interference," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2007, pp. 171–182.

[30] Y. Li, R. Mahajan, L. Qiu, E. Rozner, and Y. Zhang, "Predictable performance optimization for wireless networks," in *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*. New York, NY, USA: ACM, 2008, pp. 413–426.

[31] L. Chen, M. Chiang, J. C. Doyle, and S. H. Low, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *25th IEEE International Conference on Computer Communications*, 2006, pp. 1–13.

[32] M. Bresler, J. Chiang, M. He, , and J. Rexford, "Towards robust multi-layer traffic engineering: optimization of congestion control and routing," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 5, pp. 868–880, 2007.

[33] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 104–116, 2005.

[34] B. Bensaou and Z. Fang, "A Fair MAC Protocol for IEEE 802.11-Based Ad Hoc Networks: Design and Implementation," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 2934–2941, 2007.

[35] M. Gerla, S. Bae, and K. Xu, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks," in *IEEE Global Telecommunications Conference*, vol. 1, 2002, pp. 72–76.

[36] V. Gambiroza, E. W. Knightly, and B. Sadeghi, "End-to-end performance and fairness in multihop wireless backhaul networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2004, pp. 287–301.

[37] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless Networks*, vol. 11, no. 4, pp. 471–487, 2005.

[38] R. Mahajan, C. Reis, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 51–62, 2006.

[39] T. Moscibroda, R. Wattenhofer, and Y. Weber, "Protocol design beyond graph-based models," in *Proceedings of the 5th ACM SIGCOMM Workshop on Hot Topics in Networks*. New York, NY, USA: ACM, 2006.

[40] X. Gao, V. Bharghavan, T.-E. Kim, and T. Nandagopal, "Achieving MAC layer fairness in wireless packet networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2000, pp. 87–98.

[41] S. Low and D. Lapsley, "Optimization flow control. I. Basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[42] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 169–180, 2007.

[43] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[44] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation*. Prentice-Hall, 1989.

[45] B. Chen, J. Jannotti, M. F. Kaashoek, E. Kohler, and R. Morris, "The click modular router," *ACM Transactions on Computer Systems*, vol. 18, pp. 263–297, 2000. [Online]. Available: http://doi.acm.org/10.1145/354871.354874

[46] "Libpcap library." [Online]. Available: http://www.tcpdump.org

[47] D. Ripplinger, "Modeling wireless networks for rate control," Master's thesis, Brigham Young University, 2011.

[48] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki, "Online estimation of RF interference," in *Proceedings of the 2008 ACM CoNEXT Conference*. New York, NY, USA: ACM, 2008, pp. 1–12.

[49] N. Chiba and T. Nishizeki, "Arboricity and subgraph listing algorithms," *SIAM Journal on Computing*, vol. 14, pp. 210–223, 1985. [Online]. Available: http://dx.doi.org/10.1137/0214017

[50] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[51] "The netfilter framework and iptables homepage." [Online]. Available: http://www.netfilter.org

[52] "IEEE 802.11 working group website." [Online]. Available: http://www.ieee802.org/11

[53] A. Adya, P. Bahl, L. Zhou, J. Padhye, and A. Wolman, "A multi-radio unification protocol for IEEE 802.11 wireless networks," in *First International Conference on Broadband Networks*, 2004, pp. 344–354.

[54] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2004, pp. 114–128.

[55] A. Rai, D. Ripplinger, S. Warnick, L. Wang, and D. Zappala, "A convex optimization approach to decentralized rate control in wireless networks with partial interference," in *49th IEEE Conference on Decision and Control*. Atlanta, GA, USA: IEEE, 2010, pp. 639 – 646.